LNCS 6017

David Taniar   Osvaldo Gervasi
Beniamino Murgante   Eric Pardede
Bernady O. Apduhan (Eds.)

Computational Science and Its Applications – ICCSA 2010

International Conference
Fukuoka, Japan, March 2010
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Computer Science 6017

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

David Taniar   Osvaldo Gervasi
Beniamino Murgante   Eric Pardede
Bernady O. Apduhan (Eds.)

Computational Science and Its Applications – ICCSA 2010

International Conference
Fukuoka, Japan, March 23-26, 2010
Proceedings, Part II

Springer

Volume Editors

David Taniar
Monash University, Clayton, VIC 3800, Australia
E-mail: david.taniar@infotech.monash.edu.au

Osvaldo Gervasi
University of Perugia, 06123 Perugia, Italy
E-mail: osvaldo@unipg.it

Beniamino Murgante
University of Basilicata, L.I.S.U.T. - D.A.P.I.T., 85100 Potenza, Italy
E-mail: beniamino.murgante@unibas.it

Eric Pardede
La Trobe University, Bundoora, VIC 3083, Australia
E-mail: e.pardede@latrobe.edu.au

Bernady O. Apduhan
Kyushu Sangyo University, Fukuoka, 813-8503 Japan
E-mail: bob@is.kyusan-u.ac.jp

Library of Congress Control Number: 2010922807

CR Subject Classification (1998): C.2, H.4, F.2, H.3, C.2.4, F.1

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-642-12164-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-12164-7 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

These multiple volumes (LNCS volumes 6016, 6017, 6018 and 6019) consist of the peer-reviewed papers from the 2010 International Conference on Computational Science and Its Applications (ICCSA2010) held in Fukuoka, Japan during March 23–26, 2010. ICCSA 2010 was a successful event in the International Conferences on Computational Science and Its Applications (ICCSA) conference series, previously held in Suwon, South Korea (2009), Perugia, Italy (2008), Kuala Lumpur, Malaysia (2007), Glasgow, UK (2006), Singapore (2005), Assisi, Italy (2004), Montreal, Canada (2003), and (as ICCS) Amsterdam, The Netherlands (2002) and San Francisco, USA (2001).

Computational science is a main pillar of most of the present research, industrial and commercial activities and plays a unique role in exploiting ICT innovative technologies. The ICCSA conference series has been providing a venue to researchers and industry practitioners to discuss new ideas, to share complex problems and their solutions, and to shape new trends in computational science.

ICCSA 2010 was celebrated at the host university, Kyushu Sangyo University, Fukuoka, Japan, as part of the university's 50th anniversary. We would like to thank Kyushu Sangyo University for hosting ICCSA this year, and for including this international event in their celebrations. Also for the first time this year, ICCSA organized poster sessions that present on-going projects on various aspects of computational sciences.

Apart from the general track, ICCSA 2010 also included 30 special sessions and workshops in various areas of computational sciences, ranging from computational science technologies, to specific areas of computational sciences, such as computer graphics and virtual reality. We would like to show our appreciation to the workshops and special sessions Chairs and Co-chairs.

The success of the ICCSA conference series, in general, and ICCSA 2010, in particular, was due to the support of many people: authors, presenters, participants, keynote speakers, session Chairs, Organizing Committee members, student volunteers, Program Committee members, Steering Committee members, and people in other various roles. We would like to thank them all. We would also like to thank Springer for their continuous support in publishing ICCSA conference proceedings.

March 2010

Osvaldo Gervasi
David Taniar

Organization

ICCSA 2010 was organized by the University of Perugia (Italy), Monash University (Australia), La Trobe University (Australia), University of Basilicata (Italy), and Kyushu Sangyo University (Japan)

Honorary General Chairs

Takashi Sago	Kyushu Sangyo University, Japan
Norio Shiratori	Tohoku University, Japan
Kenneth C.J. Tan	Qontix, UK

General Chairs

Bernady O. Apduhan	Kyushu Sangyo University, Japan
Osvaldo Gervasi	University of Perugia, Italy

Advisory Committee

Marina L. Gavrilova	University of Calgary, Canada
Andrès Iglesias	University of Cantabria, Spain
Tai-Hoon Kim	Hannam University, Korea
Antonio Laganà	University of Perugia, Italy
Katsuya Matsunaga	Kyushu Sangyo University, Japan
Beniamino Murgante	University of Basilicata, Italy
Kazuo Ushijima	Kyushu Sangyo University, Japan (ret.)

Program Committee Chairs

Osvaldo Gervasi	University of Perugia, Italy
David Taniar	Monash University, Australia
Eric Pardede (Vice-Chair)	LaTrobe University, Australia

Workshop and Session Organizing Chairs

Beniamino Murgante	University of Basilicata, Italy
Eric Pardede	LaTrobe University, Australia

Publicity Chairs

Jemal Abawajy	Deakin University, Australia
Koji Okamura	Kyushu Sangyo University, Japan
Yao Feng-Hui	Tennessee State University, USA
Andrew Flahive	DSTO, Australia

International Liaison Chairs

Hiroaki Kikuchi	Tokay University, Japan
Agustinus Borgy Waluyo	Institute for InfoComm Research, Singapore
Takashi Naka	Kyushu Sangyo University, Japan

Tutorial Chair

Andrès Iglesias	University of Cantabria, Spain

Awards Chairs

Akiyo Miyazaki	Kyushu Sangyo University, Japan
Wenny Rahayu	LaTrobe University, Australia

Workshop Organizers

Application of ICT in Healthcare (AICTH 2010)

Salim Zabir	France Telecom /Orange Labs Japan
Jemal Abawajy	Deakin University, Australia

Approaches or Methods of Security Engineering (AMSE 2010)

Tai-hoon Kim	Hannam University, Korea

Advances in Web-Based Learning (AWBL 2010)

Mustafa Murat Inceoglu	Ege University (Turkey)

Brain Informatics and Its Applications (BIA 2010)

Heui Seok Lim	Korea University, Korea
Kichun Nam	Korea University, Korea

Computer Algebra Systems and Applications (CASA 2010)

Andrès Iglesias	University of Cantabria, Spain
Akemi Galvez	University of Cantabria, Spain

Computational Geometry and Applications (CGA 2010)

Marina L. Gavrilova	University of Calgary, Canada

Computer Graphics and Virtual Reality (CGVR 2010)

Osvaldo Gervasi	University of Perugia, Italy
Andrès Iglesias	University of Cantabria, Spain

Chemistry and Materials Sciences and Technologies (CMST 2010)

Antonio Laganà	University of Perugia, Italy

Future Information System Technologies and Applications (FISTA 2010)

Bernady O. Apduhan	Kyushu Sangyo University, Japan
Jianhua Ma	Hosei University, Japan
Qun Jin	Waseda University, Japan

Geographical Analysis, Urban Modeling, Spatial Statistics (GEOG-AN-MOD 2010)

Stefania Bertazzon	University of Calgary, Canada
Giuseppe Borruso	University of Trieste, Italy
Beniamino Murgante	University of Basilicata, Italy

Graph Mining and Its Applications (GMIA 2010)

Honghua Dai	Deakin University, Australia
James Liu	Hong Kong Polytechnic University, Hong Kong
Min Yao	Zhejiang University, China
Zhihai Wang	Beijing JiaoTong University, China

High-Performance Computing and Information Visualization (HPCIV 2010)

Frank Dévai	London South Bank University, UK
David Protheroe	London South Bank University, UK

International Workshop on Biomathematics, Bioinformatics and Biostatistics (IBBB 2010)

Unal Ufuktepe	Izmir University of Economics, Turkey
Andres Iglesias	University of Cantabria, Spain

International Workshop on Collective Evolutionary Systems (IWCES 2010)

Alfredo Milani	University of Perugia, Italy
Clement Leung	Hong Kong Baptist University, Hong Kong

International Workshop on Human and Information Space Symbiosis (WHISS 2010)

Takuo Suganuma	Tohoku University, Japan
Gen Kitagata	Tohoku University, Japan

Mobile Communications (MC 2010)

Hyunseung Choo	Sungkyunkwan University, Korea

Mobile Sensor and Its Applications (MSIA 2010)

Moonseong Kim	Michigan State University, USA

Numerical Methods and Modeling/Simulations in Computational Science and Engineering (NMMS 2010)

Elise de Doncker	Western Michigan University, USA
Karlis Kaugars	Western Michigan University, USA

Logical, Scientific and Computational Aspects of Pulse Phenomena in Transitions (PULSES 2010)

Carlo Cattani	University of Salerno, Italy
Cristian Toma	Corner Soft Technologies, Romania
Ming Li	East China Normal University, China

Resource Management and Scheduling for Future-Generation Computing Systems (RMS 2010)

Jemal H. Abawajy	Deakin University, Australia

Information Retrieval, Security and Innovative Applications (RSIA 2010)

Mohammad Mesbah Usddin	Kyushy University, Japan

Rough and Soft Sets Theories and Applications (RSSA 2010)

Mustafa Mat Deris	Universiti Tun Hussein Onn, Malaysia
Jemal H. Abawajy	Deakin University, Australia

Software Engineering Processes and Applications (SEPA 2010)

Sanjay Misra	Atilim University, Turkey

Tools and Techniques in Software Development Processes (TTSDP 2010)

Sanjay Misra	Atilim University, Turkey

Ubiquitous Web Systems and Intelligence (UWSI 2010)

David Taniar	Monash University, Australia
Eric Pardede	La Trobe University, Australia
Wenny Rahayu	La Trobe University, Australia

Wireless and Ad-Hoc Networking (WADNet 2010)

Jongchan Lee	Kunsan National University, Korea
Sangjoon Park	Kunsan National University, Korea

WEB 2.0 and Social Networks (Web2.0 2010)

Vidyasagar Potdar	Curtin University of Technology, Australia

Workshop on Internet Communication Security (WICS 2010)

José Maria Sierra Camara	University of Madrid, Spain

Wireless Multimedia Sensor Networks (WMSN 2010)

Vidyasagar Potdar	Curtin University of Technology, Australia
Yan Yang	Seikei University, Japan

Program Committee

Kenneth Adamson	Ulster University, UK
Margarita Albertí Wirsing	Universitat de Barcelona, Spain
Richard Barrett	Oak Ridge National Laboratory, USA
Stefania Bertazzon	University of Calgary, Canada
Michela Bertolotto	University College Dublin, Ireland
Sandro Bimonte	CEMAGREF, TSCF, France
Rod Blais	University of Calgary, Canada
Ivan Blecic	University of Sassari, Italy
Giuseppe Borruso	Università degli Studi di Trieste, Italy
Martin Buecker	Aachen University, Germany
Alfredo Buttari	CNRS-IRIT, France
Carlo Cattani	University of Salerno, Italy
Alexander Chemeris	National Technical University of Ukraine “KPI”, Ukraine
Chen-Mou Cheng	National Taiwan University, Taiwan
Min Young Chung	Sungkyunkwan University, Korea
Rosa Coluzzi	National Research Council, Italy
Stefano Cozzini	National Research Council, Italy
José A. Cardoso e Cunha	Univ. Nova de Lisboa, Portugal
Gianluca Cuomo	University of Basilicata, Italy
Alfredo Cuzzocrea	University of Calabria, Italy
Ovidiu Daescu	University of Texas at Dallas, USA
Maria Danese	University of Basilicata, Italy
Pravesh Debba	CSIR, South Africa
Oscar Delgado-Mohatar	University Carlos III of Madrid, Spain
Roberto De Lotto	University of Pavia, Italy

Jean-Cristophe Desplat	Irish Centre for High-End Computing, Ireland
Frank Dévai	London South Bank University, UK
Rodolphe Devillers	Memorial University of Newfoundland, Canada
Pasquale Di Donato	Sapienza University of Rome, Italy
Carla Dal Sasso Freitas	UFRGS, Brazil
Francesco Gabellone	National Research Council, Italy
Akemi Galvez	University of Cantabria, Spain
Marina Gavrilova	University of Calgary, Canada
Nicoletta Gazzea	ICRAM, Italy
Jerome Gensel	LSR-IMAG, France
Andrzej M. Goscinski	Deakin University, Australia
Alex Hagen-Zanker	Cambridge University, UK
Muki Haklay	University College London, UK
Hisamoto Hiyoshi	Gunma University, Japan
Choong Seon Hong	Kyung Hee University, Korea
Fermin Huarte	University of Barcelona, Spain
Andrès Iglesias	University of Cantabria, Spain
Antonio Laganà	University of Perugia, Italy
Mustafa Murat	Inceoglu Ege University, Turkey
Ken-ichi Ishida	Kyushu Sangyo University, Japan
Antonio Izquierdo	Universidad Carlos III de Madrid, Spain
Daesik Jang	Kunsan University, Korea
Peter Jimack	University of Leeds, UK
Korhan Karabulut	Yasar University, Turkey
Farid Karimipour	Vienna University of Technology, Austria
Baris Kazar	Oracle Corp., USA
Dong Seong Kim	Duke University, USA
Pan Koo Kim	Chosun University, Korea
Ivana Kolingerova	University of West Bohemia, Czech Republic
Dieter Kranzlmueller	Ludwig Maximilians University and Leibniz Supercomputing Centre Munich, Germany
Domenico Labbate	University of Basilicata, Italy
Rosa Lasaponara	National Research Council, Italy
Maurizio Lazzari	National Research Council, Italy
Xuan Hung Le	University of South Florida, USA
Sangyoun Lee	Yonsei University, Korea
Bogdan Lesyng	Warsaw University, Poland
Clement Leung	Hong Kong Baptist University, Hong Kong
Chendong Li	University of Connecticut, USA
Laurence Liew	Platform Computing, Singapore
Xin Liu	University of Calgary, Canada
Cherry Liu Fang	U.S. DOE Ames Laboratory, USA
Savino Longo	University of Bari, Italy
Tinghuai Ma	NanJing University of Information Science and Technology, China
Antonino Marvuglia	University College Cork, Ireland

Michael Mascagni	Florida State University, USA
Nikolai Medvedev	Institute of Chemical Kinetics and Combustion SB RAS, Russia
Nirvana Meratnia	University of Twente, The Netherlands
Alfredo Milani	University of Perugia, Italy
Sanjay Misra	Atilim University, Turkey
Asish Mukhopadhyay	University of Windsor, Canada
Beniamino Murgante	University of Basilicata, Italy
Takashi Naka	Kyushu Sangyo University, Japan
Jiri Nedoma	Academy of Sciences of the Czech Republic, Czech Republic
Laszlo Neumann	University of GironAlexey Rodionova, Spain
Belen Palop	Universidad de Valladolid, Spain
Dimos N. Pantazis	Technological Educational Institution of Athens, Greece
Luca Paolino	Università di Salerno, Italy
Marcin Paprzycki	Polish Academy of Sciences, Poland
Gyung-Leen Park	Cheju National University, Korea
Kwangjin Park	Wonkwang University, Korea
Paola Perchinunno	University of Bari, Italy
Carlo Petrongolo	University of Siena, Italy
Antonino Polimeno	University of Padova, Italy
Jacynthe Pouliot	Université Laval, France
David C. Prosperi	Florida Atlantic University, USA
Dave Protheroe	London South Bank University, UK
Richard Ramaroso	Harvard University, USA
Jerzy Respondek	Silesian University of Technology, Poland
Alexey Rodionov	Institute of Computational Mathematics and Mathematical Geophysics, Russia
Jon Rokne	University of Calgary, Canada
Octavio Roncero	CSIC, Spain
Maytham Safar	Kuwait University, Kuwait
Haiduke Sarafian	The Pennsylvania State University, USA
Bianca Schön	University College Dublin, Ireland
Qi Shi	Liverpool John Moores University, UK
Dale Shires	U.S. Army Research Laboratory, USA
Olga Sourina	Nanyang Technological University, Singapore
Henning Sten	Copenhagen Institute of Technology, Denmark
Kokichi Sugihara	Meiji University, Japan
Francesco Tarantelli	University of Perugia, Italy
Jesús Téllez	Universidad Carlos III de Madrid, Spain
Parimala Thulasiraman	University of Manitoba, Canada
Giuseppe A. Trunfio	University of Sassari, Italy
Mario Valle	Swiss National Supercomputing Centre, Switzerland

Pablo Vanegas	Katholieke Universiteit Leuven, Belgium
Piero Giorgio Verdini	INFN Pisa and CERN, Italy
Andrea Vittadini	University of Padova, Italy
Koichi Wada	University of Tsukuba, Japan
Krzysztof Walkowiak	Wroclaw University of Technology, Poland
Jerzy Wasniewski	Technical University of Denmark, Denmark
Robert Weibel	University of Zurich, Switzerland
Roland Wismüller	Universität Siegen, Germany
Markus Wolff	University of Potsdam, Germany
Kwai Wong	University of Tennessee, USA
Mudasser Wyne	National University, USA
Chung-Huang Yang	National Kaohsiung Normal University, Taiwan
Albert Y. Zomaya	University of Sydney, Australia

Sponsoring Organizations

ICCSA 2010 would not have been possible without the tremendous support of many organizations and institutions, for which all organizers and participants of ICCSA 2010 express their sincere gratitude:

University of Perugia, Italy
Kyushu Sangyo University, Japan
Monash University, Australia
La Trobe University, Australia
University of Basilicata, Italia
Information Processing Society of Japan (IPSJ) - Kyushu Chapter
and with IPSJ SIG-DPS

Table of Contents – Part II

Workshop on Numerical Methods and Modeling/Simulations in Computational Science and Engineering (NMMS 2010)

Workshop on PULSES V- Logical, Scientific and Computational Aspects of Pulse Phenomena in Transitions (PULSES 2010)

Workshop on Software Engineering Processes and Applications (SEPA 2010)

Workshop on WEB 2.0 and Social Networks (Web2.0 2010)

General Track on Information Systems and Technologies

General Track on Computational Methods, Algorithms and Scientific Application

General Track on Advanced and Emerging Applications

The Use of Shadow Regions in Multi Region FDM: High Precision Cylindrically Symmetric Electrostatics

David Edwards, Jr.

IJL Research Center
Newark, VT 05871 USA
802 467 1177
dej@kingcon.com

Abstract. A previously reported multi region FDM process, while producing very accurate solutions to the cylindrically symmetric electrostatic problem for points distant from region boundaries has been found to have abnormally high errors near the region boundaries themselves. A solution to this problem has been found that uses an extension of the definition of a region boundary line to a two line array. The resultant regions are called shadow regions and it is shown that their use in the multi region FDM process reduces the previous errors by ~ two orders of magnitude thus significantly extending the precisional capabilities of the multi region FDM method.

Keywords: FDM, High Precision Calculations, Multi Region FDM.

1 Introduction

Cylindrically symmetric electrostatics is an area that involves the calculation of electrostatic potentials in problems in which the potential satisfies Laplace's equation interior to a closed geometry. This finds application in lens design for charged particles in which one uses the potentials within the region to trace particle trajectories through the lens. In a number of such applications high precision trajectories are required necessitating high accuracy potential calculations as the accuracy of the trajectory is related to the precision of the potentials. The high accuracy potential calculation has been approached using a variety of techniques [1, 2] being reviewed in a useful monograph by Heddle [3].

One of the most widely used and conceptually simple techniques for solving this type of problem is the single region finite difference method (FDM). To improve the accuracy of this method from its nominal precision value of ~10^{-6}, the multi region FDM method was created in 1983 [4]. This was seen to greatly enhance the precision capabilities of the single region FDM method while retraining the simplicity of the single region process itself. Since that initial work, several developments have occurred extending its precision capabilities:

i. The construction and use of very high order algorithms [5, 6].
ii. The calculation of potentials one unit from a metal surface [7].
iii. Auto establishing the multi region structure based on a precision desired [6].

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 1–13, 2010.

It was found with the above improvements that mesh point values far from any region boundaries were reasonably accurate [8] (maximum on axis error of ~2*10^{-15}). However, it has been observed that for mesh points in the vicinity of a region boundary, abnormally high errors occur. It is the determination of the cause of these errors and their subsequent reduction that is the subject of this paper.

1.1 Background

To understand the multi region technique brief descriptions of both the single region FDM and multi region FDM process will be given. This will enable the cause of the problem to be inferred and its solution to be implemented.

1.2 The Geometry

The geometry used to create the precisional problem near a region boundary will be the two tube geometry consisting of two coaxial cylinders brought together with zero gap and closed at each end. This geometry is called tube10 and is shown in figure 1.

In this figure the two coaxial cylinders are shown joined together along a common axis with the left hand cylinder having a potential of 0 and the right hand cylinder a potential of 10. This will provide an example appropriate to the multi region technique in which there is a localized region (near the point of discontinuity) having

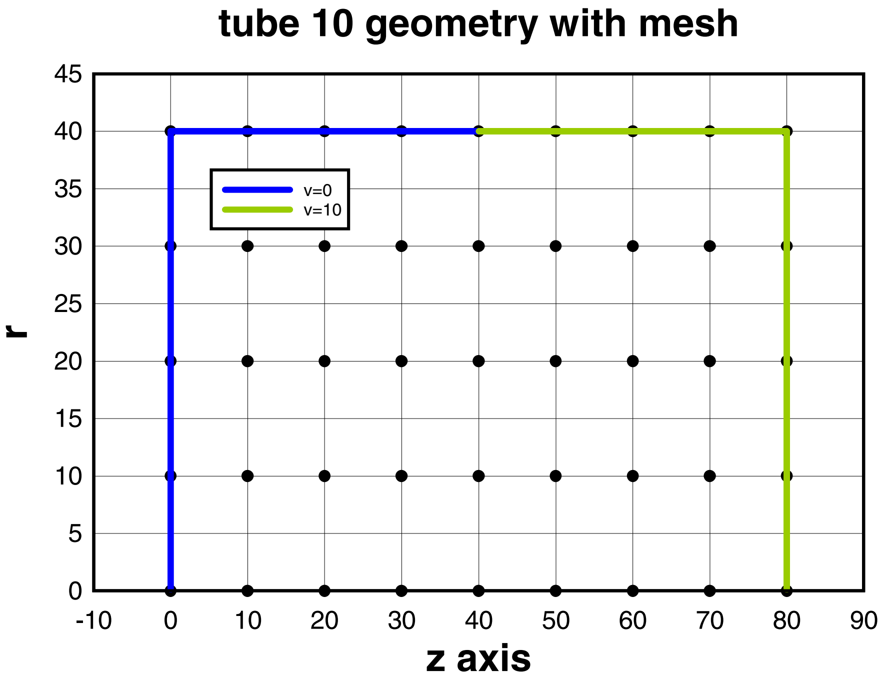


Fig. 1. Seen is the tube 10 geometry consisting of two coaxial cylinders brought together with zero gap. Representative mesh points are displayed.

rapidly varying fields. One of the properties of this geometry is that it has a known analytic solution [9] allowing errors resulting from any calculation to be determined.

1.3 Single Region FDM

As the multi region FDM process relies heavily on the single region FDM process in both terminology and technique, the following is a brief description of the single region process. Since the geometry is contained within a closed surface on which the potential is specified, the problem is known to have a well defined solution.

After placing a uniform mesh is over the geometry the process begins by setting the values of the mesh points on the boundary to those of the geometry and initializing the remainder (typically to 0). One then steps through the mesh and at each mesh point calculates its value using only the values of neighboring points. The above iteration is continued until the maximum change in the value at any point during the iteration is less than an end criterion value. This process is termed a *relaxation process* and at the termination the net is said to be *relaxed.*

1.4 Algorithms

From the above it is seen that an algorithm is required to determine the value at any mesh point from the values of its neighboring points. The algorithm construction process has been previously described [4, 5] and only a brief review is given here.

It proceeds in the following manner. For the sake of definitess the process for a 10th order algorithm will be described as it is the algorithm that is utilized for general points [4] within the net.

The potential in a neighborhood of any mesh point is expanded in terms of a 10th order power series in r, z, the coordinates being relative to the mesh point itself.

$$v(r,z) = c_0+c_1z+c_2r+c_3z^2 +c_4zr+c_5r^2 + ... + c_jz^{10} +...+c_{66}r^{10} + O(zr^{11}). \qquad (1)$$

where O (zr^{11}) means that terms having factors r^jz^k are neglected for j+k=11 and above.

There are 67 coefficients (c_0 … c_{66}) to be determined. Applying Laplace's equation to (1) and requiring that the resulting equation be valid in an arbitrary neighborhood of the chosen mesh point results in 45 equations. Thus an additional 22 equations are required to solve for the complete set of c_j's and are found by evaluating (1) at a selection of 22 neighboring mesh points. (The number is a function of the order of the algorithm.) Although the particular choice of selected mesh points is somewhat arbitrary, it has been found that the precision of the resultant algorithm is maximized by choosing mesh points both close to and symmetric about the central mesh point. Discussions of the above can be found in references [4, 5, and 8].

It is noted that the above process produces solutions for all c_j's. While the potential at the central point is given simply by c_0, the determination of the complete set $\{c_j\}$ will enable interpolations for the potential values at ½ integral values to be made with interpolated precisions approximately those of c_0.

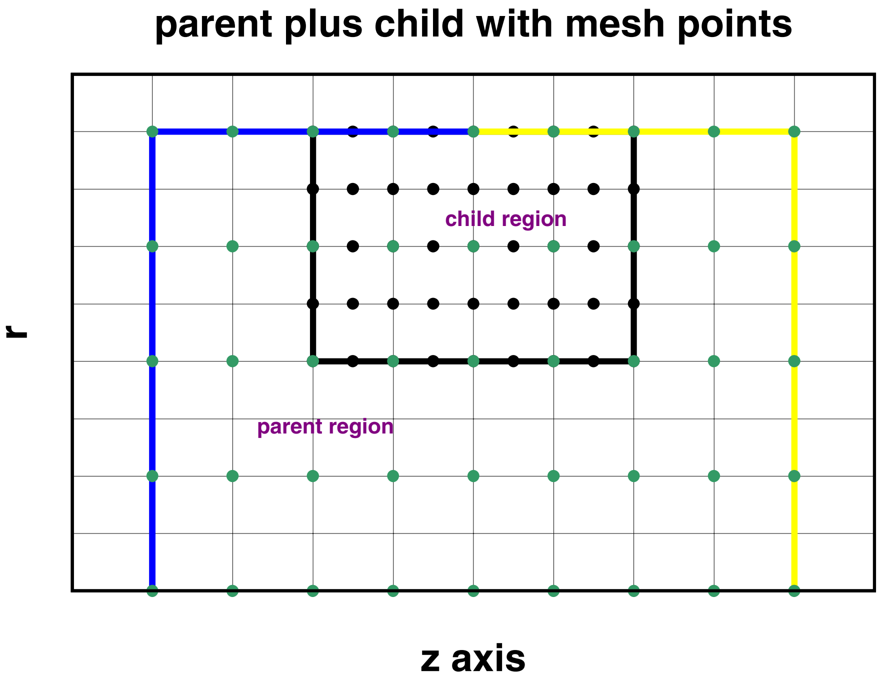


Fig. 2. A child region is shown as a sub region of the parent containing both the points of the parent and points lying at ½ integral values of the parent

1.5 Multi Region FDM Process

A net is placed over the geometry as described in section 1.3 and is termed the main net. This net is considered the parent for any immediately contained sub regions which are called child regions of the parent. (A child region is thus the child of a parent and the parent of its children.) The rectangular boundary of a child is constructed so its corners lie on parent mesh points. The child mesh points are then placed within the region at half integral values of the parent giving a density enhancement of a factor of 4 over that of its parent. Figure 2 illustrates such a construction.

The illustrated two region structure is relaxed in the following manner:

i. The parent is considered a single region with closed boundary and is relaxed through n cycles described previously.

ii. The mesh points on the boundary lines of the child are set from the recently determined parent values, interpolating the child values at the ½ integral boundary points using (1).

iii. The child mesh is then considered a single region mesh having its boundary specified and it is relaxed n cycles. After this relaxation the parent points that have their images in the child are set from the corresponding child values.

iv. When relaxing the parent mesh no parent point contained within the child is relaxed as it will be subsequently relaxed with greater precision when the child is relaxed.

v. The above is continued until an end criterion is reached.

The relaxation of a multi region structure is easily generalized from this two region relaxation description.

1.6 The Grad6 Function

A function called the grad6 is defined [5] as rms value of the coefficients of the terms in the power series expansion (1) having factors r^jz^k where j+k=6. As noted above since all of the required coefficients can be determined, grad6 may be found at any mesh point. As every point has a value of grad6 and an error value, the error at a mesh point can be plotted vs its grad6 value, an example of which will be given in figure 9.

1.7 Multi Region Structure Determinations

The auto creation of a multi region structure having a desired precision specified at the outset has been previously reported [6]. The essential features of the process are: From the desired precision a quantity called "cut" is determined with the property that if the grad6 function evaluated at a mesh point is less than "cut" its algorithmic error

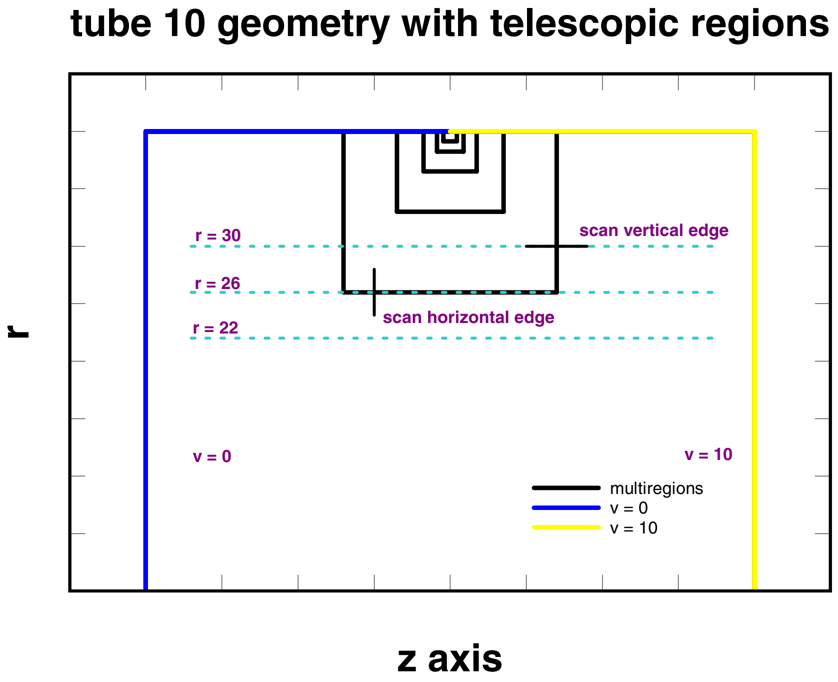


Fig. 3. The multi region structure for the tube 10 geometry is shown. Also shown are dashed lines at various r values used in subsequent error scans.

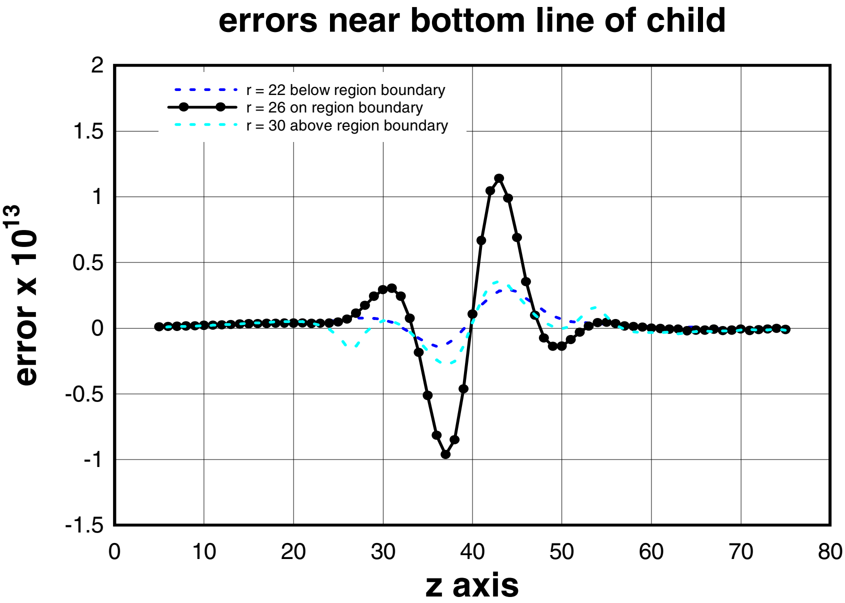
will be less than the desired precision. Thus after a region is relaxed, grad6 values for all of the mesh points in the region are found and a child region is placed around all of those mesh points with grad6 values greater than cut. In this way the region is partitioned into two disjoint regions, the child containing the lower precision values. This structure is then relaxed as described above and the child is then treated as a parent, its grad6 values found and used to create its child. This process is continued until a predetermined total number of regions has been created at which time the entire structure is relaxed with a possibly more stringent end criterion. It is noted that in the final relax, a high precision arithmetic software unit has been used in place of the built in microcode of the CPU. The software unit had an lsb of ~10^{-30}.

Applying the process to our example with the desired precision set at ~$5x10^{-16}$ has resulted in the following region structure.

Seen is both the symmetric nature of the child regions about the point of potential discontinuity and their convergence to the point of discontinuity itself.

2 Errors Near a Region Boundary

Using the relaxation process described above the values within the net can be established and the error at each mesh point determined (using the exact solution as the reference). Horizontal error scans may then be made near the bottom of the first child region at r values of 22, 26 and 30 as shown in the above figure. The results of these scans are given in figure 4 in which it is seen that the largest error occurs on the



F

Fig. 4. The errors along 3 horizontal lines located near the bottom of the child region are plotted showing that the error is highest on the region boundary decreasing for scan lines either above or below the child boundary line

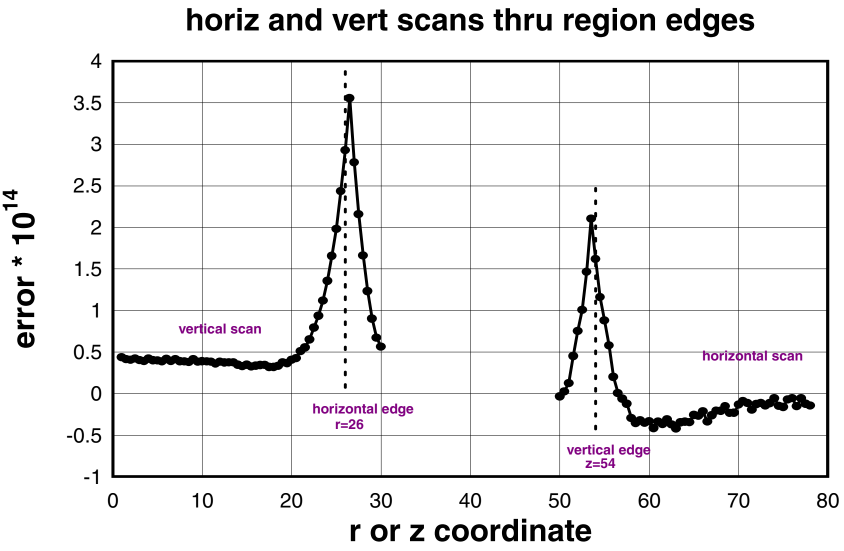


Fig. 5. The errors along the shorter scan lines in figure 3 show that the errors likely originate from the algorithms used for points either on or quite near the boundary lines of the region

boundary line itself decreasing for the scans either above or below this line suggesting that algorithms used near the boundary itself are causing the large errors on the boundary. This effect is also seen from scans perpendicular to a horizontal and vertical boundary line of the child. The plot of the errors along the scan lines is given in figure 5. (The short scan lines are shown in figure 3 being named “scan horizontal edge” and “scan vertical edge”.)

From these results it is clear that the algorithmic determination of the value at mesh points on or near a region boundary has large associated errors. There are 2 possible causes for these errors: The first and perhaps most obvious is that the algorithm for setting the potential on the region boundary itself is the problem. However since only ½ of the mesh points on the boundary require an interpolation, and since the imprecision of the interpolation algorithm (described in a later section) is less than a factor of 10 general mesh point precision, the resultant imprecision near the region boundary can be estimated to be ~ factor of 5 of that of the general mesh point algorithm which is too small to explain the above observations.

Hence the most likely cause for the observed errors occurs not from setting the boundary potentials of the child but from the algorithm used during the relaxation process for mesh points one unit from a boundary.

The algorithm used for mesh points one unit from a boundary was not the general order 10 algorithm used in the rest of the net. The reason for this can be seen from figure 6 in which the required mesh points for the order 10 algorithm are shown for a mesh point one unit to the right of the left boundary line of the child. Seen is that were one to use the 10th order algorithm, two mesh points required for this algorithm would be outside the child region and hence not available. One might consider keeping the

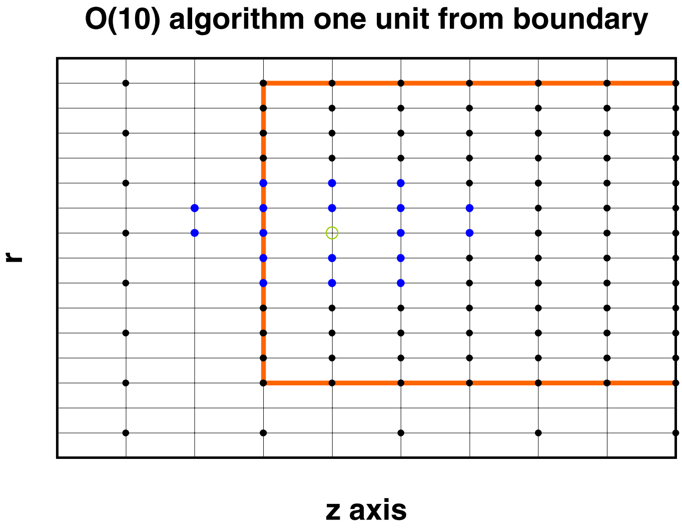


Fig. 6. A child region is shown embedded in its less dense parent together with the mesh points required for a general order 10 algorithm. The central point for the algorithm is the open circle one unit from a boundary line.

order 10 algorithm for mesh points one unit from a boundary by using an unsymmetrical set of meshpoints. Although it is possible to create such algorithms and the determined algorithms have in fact precisions only slightly degraded from the general mesh point algorithm, *the relaxation process using these algorithms was found to be unstable.*

Due to this necessary restriction the algorithm that was finally chosen for use in the relaxation process was a 6th order algorithm. Use of this algorithm would result in degradation in precision over that of the 10th order algorithm [5] which would only occur for the points near a region boundary as these were the only points using this lower order algorithm.

In view of the above a new approach to the region boundary needed to be found which would overcome the need for the order 6 algorithm for these points.

3 The Shadow Region

It was decided to generalize the concept of a boundary line to a two line boundary array contained strictly within the defining boundary lines of the child region. The points in the two line array are called *shadow points* and the region so constructed is called a *shadow region.* The geometry of the shadow points in the shadow region is shown in figure 7.

The values at the shadow points themselves are determined from the parent mesh by the interpolation methods described above and a 10th order algorithm can be used. The relaxation process itself is the same as described for non shadow regions with the

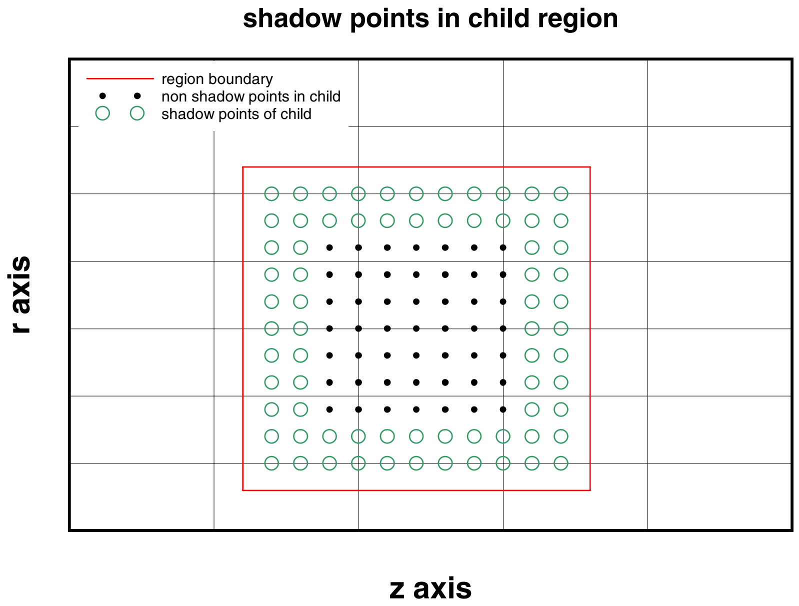


Fig. 7. Shadow points (larger open circles) defining the boundary of the child are shown along with the non shadow points of the child (smaller discs)

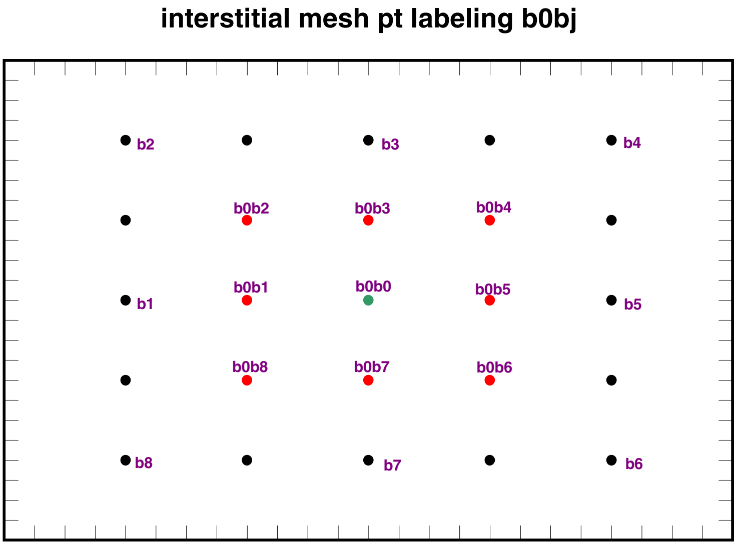


Fig. 8. The complete set of interstitial meshpoints b0bj surrounding a parent point at b0b0 are shown. b0 … b8 are the parent points surrounding the central point b0b0

understanding that shadow points are to be considered boundary points and hence not relaxed when relaxing a region. *The immediate implication of the above construction is that the 10th order algorithm can be used for all non shadow points within the region during the relaxation process since these points have the required neighboring mesh points.*

3.1 10th Order Interpolation Algorithms

The ½ integral points denoted by b0bj and the integral mesh points denoted by b_j are defined in the following figure.

The order 10 algorithm construction for the interstitial mesh points has been described elsewhere [5, 6]. The error at any mesh point in tube 10 may be plotted vs its grad6 value and the results are given in figure 9. (Only plots for b0b1,…,b0b5 are displayed since only these are required to determine the values at any of the shadow points.)

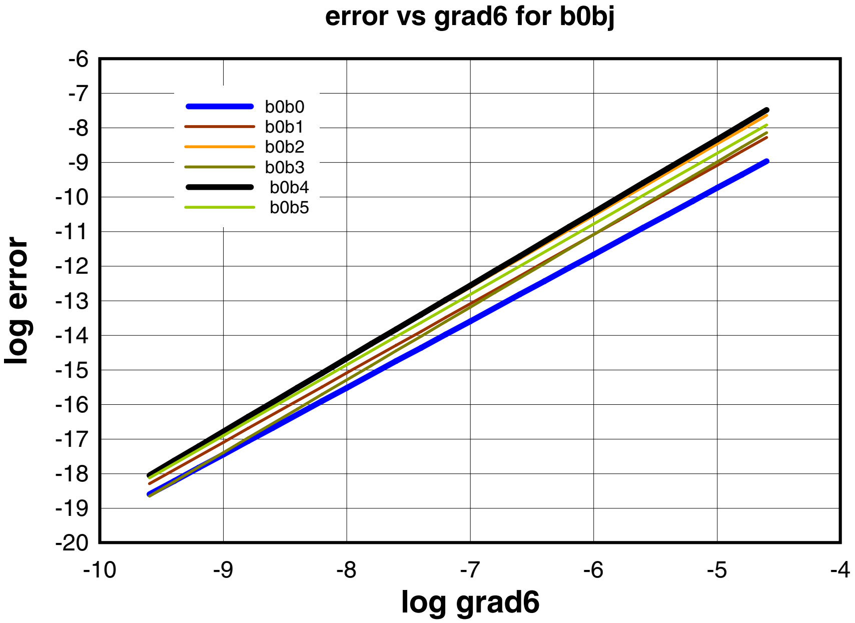


Fig. 9. The error vs grad6 is shown for 10th order interpolation algorithms for interstitial points b0b0 … b0b5. It is noted that every mesh point within the geometry is represented by a point on the log grad6 axis.

Seen is that while the 10th order interpolation algorithm can be a factor of ~ 10 less precise than general mesh point algorithm (b0b0) its imprecision is considerably less than the precision degradation of the values of mesh points near a region boundary (figures 4 and 5).

4 Results

As seen from figures 4 and 5 the maximum error in the main net occurs on a horizontal scan line very near the child boundary. To test the efficacy of the shadow region solution, two nets were constructed having the same geometric multi region structure, one using non shadow regions and the other shadow. For these nets horizontal scans were made one unit above the first child boundary. The results are shown in figure 10.

Seen from the above figure is that the use of shadow regions has reduced the error near the region boundary by ~two orders of magnitude over the non shadow net thus supporting the supposition that the observed errors of figures 4 and 5 were due to the restrictions on the order of the algorithm able to be used one unit from a boundary. It may be worthwhile to mention that this reduction is significant if the full potential of the technique is to be realized.

And finally in figure 11 horizontal error scans both on axis and at r =30 are plotted for tube 10 using shadow regions. It is noted that the r=30 scan is an upper bound to the error for any r value less than 30. In this figure we see that one is able to achieve precisions of ~$5*10^{-17}$ for a large fraction of the geometry and an on axis error of <$1.5*10^{-17}$.

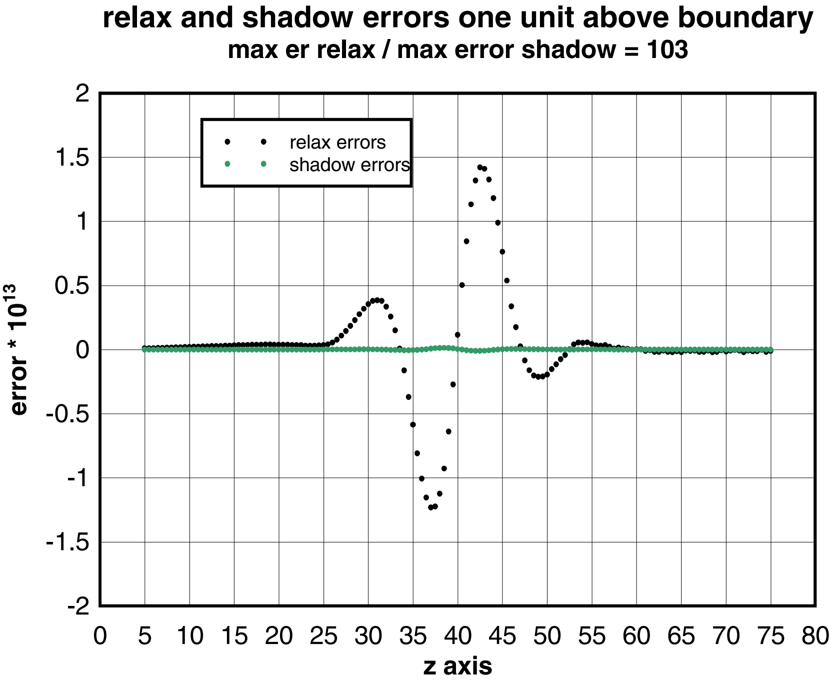


Fig. 10. A comparison of the shadow and non shadow error plots on a scan line one unit above the first child boundary. Seen is that the use of shadow regions has reduced the error near the region boundary by ~ two orders of magnitude.

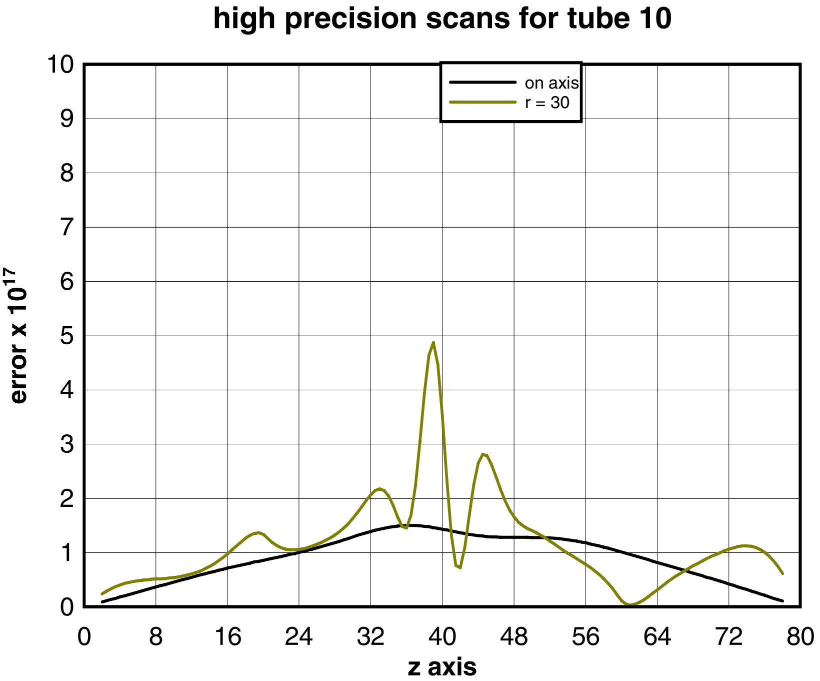


Fig. 11. Plotted are the resultant errors for tube 10 using shadow regions for a scan both on axis and at r=30, the r=30 scan being an upper error bound for any r<30

5 Conclusion

The construction of shadow regions in the context of multi region FDM has effected a reduction of ~100 of the errors present near the region boundaries. This has enabled a multi region calculation to achieve precisions ~< $5*10^{-17}$ over ¾ of the geometrical area.

References

1. Read, F.H., Adams, A., Soto-Montiel, J.R.: Electrostatic Cylinder Lenses I: Two Element Lenses. J. Phys. E 4, 625 (1971)
2. Natali, S., DiChiro, D., Kuyatt, C.E.: Accurate Calculations of Properties of the Two Tube Electrostatic Lens I. Improved Digital Methods for the Precise Calculation of Electric Fields and Trajectories. J. Res. Nat. Bur. Stand. Sect. A 76, 27 (1972)
3. Heddle, D.W.O.: Electrostatic Lens Systems., 2nd ed. Institute of Physics Publishing, ISBN 0-7503-0697-1
4. Edwards Jr., D.: Accurate Calculations of Electrostatic Potentials for Cylindrically Symmetric Lenses. Review of Scientific Instruments 54, 1229–1235 (1983)
5. Edwards Jr., D.: High Precision Electrostatic Potential Calculations For Cylindrically Symmetric Lenses. Review of Scientific Instruments 78, 1–11 (2007)

6. Edwards Jr., D.: High precision multiregion FDM calculation of electrostatic potential. In: Advances in Industrial Engineering and Operations Research. Springer, Heidelberg (2008)
7. Edwards Jr., D.: Single Point FDM Algorithm Development for Points One Unit from a Metal Surface. In: Proceedings of International Multi Conference of Engineers and Computer Scientists 2008, Hong Kong, March 19-21 (2008)
8. Edwards Jr., D.: Accurate potential calculations for the two tube electrostatic lens using a multiregion FDM method. In: Proceedings EUROCON 2007, Warsaw, September 9-13 (2007)
9. Jackson, J.D.: Classical Electrodynamics. John Wiley & Sons, New York (1963); Library of Congress Card Number 62-8744

Fast Forth Power and Its Application in Inversion Computation for a Special Class of Trinomials

Yin Li[1], Gong-liang Chen[1], and Jian-hua Li[1,2]

[1] School of Information Security Engineering
[2] Department of Electronic Engineering,
Shanghai Jiaotong University, Shanghai, P.R. China

Abstract. This contribution is concerned with an improvement of Itoh and Tsujii's algorithm for inversion in finite field $GF(2^m)$ using polynomial basis. Unlike the standard version of this algorithm, the proposed algorithm uses forth power and multiplication as main operations. When the field is generated with a special class of irreducible trinomials, an analytical form for fast bit-parallel forth power operation is presented. The proposal can save $1T_X$ compared with the classic approach, where T_X is the delay of one 2-input XOR gate. Based on this result, the proposed algorithm for inversion achieves even faster performance, roughly improves the delay by $\frac{m}{2}T_X$, at the cost of slight increase in the space complexity compared with the standard version. To the best of our knowledge, this is the first work that proposes the use of forth power in computation of multiplicative inverse using polynomial basis and shows that it can be efficient.

Keywords: Multiplicative inverse, Itoh-Tsujii algorithm, forth power.

1 Introduction

Finite field $GF(2^m)$ is used in many areas such as error correcting codes and cryptography. In these applications, it is crucial to carry out field arithmetic operations, consist of addition, multiplication and inversion, efficiently. Inversion operation is usually required in an Elliptic Curve Cryptosystem [1] when computing point multiplication. Nevertheless, inversion computation is much more time-consuming than other operations in the field and many researchers attempt to perform this operation fast.

The inverse of a nonzero element $\alpha \in GF(2^m)$ is defined as follows: there exists an unique element $\alpha^{-1} \in GF(2^m)$ such that $\alpha^{-1} \cdot \alpha = 1$. Several algorithms [3,4,5,6,8,10]have been proposed for multiplicative inversion and some of them[6,10] are based on Fermat's theorem. Fermat's theorem implies that, since $\alpha \in GF(2^m)$ is nonzero, $\alpha^{-1} = \alpha^{2^m-2}$. Hence, inversion can be carried out by means of exponentiation by $2^m - 2$. The direct algorithm [10] is to carry out such exponentiation by iterative squaring and multiplication, which requires $m-1$ squaring operations and $m-2$ multiplications. Itoh-Tsujj algorithm (ITA)

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 14–24, 2010.

[6] reduces the number of multiplications to $O(\log_2 m)$. The original algorithm was proposed to be applied in $GF(2^m)$ with normal basis (NB) representation. Wu [7] shows that when $GF(2^m)$ is generated with an irreducible trinomial, the space complexity of Itoh-Tsujj algorithm using polynomial basis (PB) is at least as good as that using normal basis. Rodríguez-Henríquez et al. [9] proposed a novel parallel version of Itoh-Tsujii algorithm in PB using field multiplication, field squaring, and field square root operators as main building blocks.

Generally speaking, Itoh-Tsujii algorithm using PB, referred to as ITA-PB, requires more time delay compared with Itoh-Tsujii algorithm using NB in bit-parallel implementation. The main reason is that squaring in NB is given in terms of a cyclic shift of the coefficients and does not cost any gates delay while squaring in PB costs at least $1T_X$ gates delay[7](T_X denotes propagation delays of an XOR gate). Although the multipliers in two bases have nearly the same time complexity [11,12], the advantage with respect to squaring in NB leads to the less time delay for ITA. Rodríguez-Henríquez et al. approach can perform even faster but it needs a multiplexer for different blocks switching and requires a little more complicated architecture.

Our work is devoted to speedup ITA-PB at the cost of slight increase in space complexity. Unlike the standard ITA-PB, our approach is mainly based on field multiplications and field forth power operations. Consider a special class of irreducible trinomials, namely, $x^m + x^t + 1$, with m, t being odd or m being even and t being odd. In the field generated with these trinomials, squaring costs $2T_X$ gates delay [7]. The classic approach computed forth power by means of iterating squaring operation two times. Nevertheless, this approach requires $4T_X$ gates delay which is not yet efficient.

In this paper, we derive a novel approach for computation of forth power which saves $1T_x$ gates delay by means of combining two iteration together. This result can be subsequently used in optimization of inversion computation. We suggest that the proposed ITA-PB using forth power can achieve faster performance than standard ITA-PB with a speedup of about $\frac{m}{2}T_x$ gates delay, while the space complexity increase by $2m$ XOR gates at most.

The rest of this paper is organized as follows: in section 2, we briefly review the Itoh-Tsujj algorithm and some relevant concepts. Then, a bit parallel implementation of forth power operation is discussed in section 3. Based on this operation, a new novel formulation of inversion computation is presented. In section 5, comparison of our results to previous proposal for the same class of fields is made. Finally, some conclusions are drawn.

2 Itoh-Tsujii Algorithm Based on Polynomial Basis (ITA-PB)

Let $f(x) = x^m + x^t + 1$ be an irreducible trinomial over $\mathbb{F}_2$ where $m > 2t$. Then $f(x)$ induces a polynomial basis (PB) $\{1, x, x^2, \cdots, x^{m-1}\}$ in $GF(2^m)$, where x is a root of $f(x)$. In the finite field $GF(2^m)$ defined by $f(x)$, any element A represented using polynomial basis is given by:

$$A = \sum_{i=0}^{m-1} a_i x^i = a_0 + a_1 x + \cdots + a_{m-1} x^{m-1}.$$

The multiplication in PB is obtained by multiplying two polynomials and then reducing the result modulo $f(x)$. The field squaring is a special case of field multiplication:

$$C = A^2 \bmod f(x) = a_0 + a_1 x^2 + a_2 x^4 + \cdots + a_{m-1} x^{2m-2} \bmod f(x).$$

From the above expression, it is easy to see that the main operation of squaring is the reduction modulo $f(x)$. Since $f(x)$ is an irreducible trinomial, the gates count and the critical path of the circuit depend on the parity of m and t.

Since the multiplicative group of the finite field $GF(2^m)$ is cyclic of order $2^m - 1$, for any nonzero element $\alpha \in GF(2^m)$, we have $\alpha^{-1} = a^{2^m-2}$. So inversion computation consists of a power evaluation, given that:

$$\alpha^{-1} = \left(\alpha^{2^{m-1}-1}\right)^2 = \sum_{j=1}^{m-1} \alpha^{2^j}.$$

Based on this fact, we can directly obtain the result by iterative squarings and multiplications. It requires $m-2$ multiplications and $m-1$ squaring operations.

Itoh-Tsujj algorithm [6] reduced the number of required multiplications to $O(\log_2 m)$. The algorithm utilized the fact as follows:

First assume that $m - 1 = \sum_{u=1}^{n} 2^{k_u}$ with $k_1 > k_2 > \cdots > k_n$. Then consider the computation of α^{s_k} where $s_k = \sum_{i=1}^{2^k} 2^i = 2 + 2^2 + \cdots + 2^{2^k-1} + 2^{2^k}$. Note that $\alpha^{s_k} = (\alpha^{s_{k-1}})^{2^{2^{k-1}}} \cdot \alpha^{s_{k-1}}$. In computing α^{s_k}, we have also computed α^{s_i} for $s_i < s_k$. According to form of $m-1$, the multiplicative inverse of α can be rewritten as follows:

$$\alpha^{2^{m-1}+\cdots+2^2+2} = (\alpha^{s_{k_n}}) \left(\cdots (\alpha^{s_{k_3}})((\alpha^{s_{k_2}})(\alpha^{s_{k_1}})^{2^{2^{k_2}}})^{2^{2^{k_3}}} \cdots \right)^{2^{2^{k_n}}} . \quad (1)$$

Since $k_1 > k_i$ for $i = 2, \cdots, n$, then if we compute $\alpha^{s_{k_1}}$ as above, all the $\alpha^{s_{k_i}}$ for $i = 2, \cdots, t$ will also be computed. From our previous results we see that the computation of $\alpha^{s_{k_1}}$ costs $\lfloor \log_2(m-1) \rfloor$ multiplications and 2^{k_1} squaring operations. The computation of Eq. (1) totally costs $HW(m-1)-1$ multiplications and $2^{k_2} + 2^{k_3} + \cdots + 2^{k_n}$ squaring operations, where $HW(m-1)$ represents the Hamming weight of $m-1$. Adding up the partial complexities, the whole complexity of the ITA-PB costs about $\lfloor \log_2(m-1) \rfloor + HW(m-1) - 1$ field multiplications plus a total of $m-1$ squaring operations.

3 Fast Forth Power

Assume that an arbitrary element $A = \sum_{i=0}^{m-1} a_i x^i \in GF(2^m)$, the forth power of A is given by:

$$D = \sum_{i=0}^{m-1} d_i x^i = A^4 \bmod f(x) = \sum_{i=0}^{m-1} a_i x^{4i} \bmod x^m + x^t + 1$$

Actually, since the degree of A^4 is $4m-4$, the reduction operation modulo $f(x)$ is more complicated than that of degree less than $2m$, the most common method is to iterate the squaring operation two times. Then the critical path delay of corresponding circuit is twice as long as the squaring. Nevertheless, forth power based on this approach does not accelerate the ITA-PB. In this section, we will describe a new method to speedup implementation of forth power for some special cases of $f(x)$.

Explicit formulas for field squaring have been investigated in [7], it shows that when m is even and t is odd or m, t are odd, the critical path delay of squaring takes time $2T_X$. Then forth power using previous approach will cost $4T_X$ critical path delay, which is not yet efficient. Our approach is focused on the two cases. For simplicity, the case of $f(x) = x^m + x^t + 1$, with $m,\ t$ being odd numbers, is specially considered and we assume that $m,\ t$ are odd thereafter.

3.1 New Approach

Our strategy is to design an appropriate circuit for forth power and takes inspiration from the formula of squaring operation. Let $C = \sum_{i=0}^{m-1} c_i x^i = A^2 \bmod f(x)$, then

$$\sum_{i=0}^{m-1} c_i x^i = \sum_{i=0}^{m-1} a_i x^{2i} = \sum_{i=0}^{2m-2} a'_i x^i, \tag{2}$$

where a'_i is given by

$$a'_i = \begin{cases} a_{\frac{i}{2}} & \text{if } i \text{ even} \\ 0 & \text{otherwise} . \end{cases}$$

Then the coefficients c_i of C can be computed as follows[7]:

$m,\ t$ odd

$$\begin{array}{ll} c_i = a'_i, & i = 0, 2, \cdots, t-1, \\ c_i = a'_{m+i} + a'_{2m-t+i}, & i = 1, 3, \cdots, t-2, \\ c_i = a'_i + a'_{m-t+i} + a'_{2m-2t+i}, & i = t+1, t+3, \cdots, 2t-2, \\ c_i = a'_{m+i}, & i = t, t+2, \cdots, m-2, \\ c_i = a'_i + a'_{m-t+i}, & i = 2t, 2t+2, \cdots, m-1. \end{array} \tag{3}$$

According to the above expressions, note that only the third part of Eq. (3) has three operands which requires $2T_x$ critical path delay. In the implementation of forth power, we combine two squaring operations and merge the redundant operations. Namely, transfer operands $\{a'_{t+1}, a'_{t+3}, \cdots, a'_{2t-2}\}$ in Eq. (3) from the first squaring to the second squaring. If the parameter m, t satisfy certain conditions, the second squaring operation plus redundant additions would also be carried out in $2T_X$ gates delay. As a result, the whole circuit will cost $3T_X$ gates delay which can save $1T_X$ compared with previous approach.

We first compute intermediate values:

$$\begin{array}{ll} \gamma_i = a'_i, & i = 0, 2, \cdots, t-1, \\ \gamma_i = a'_{m+i} + a'_{2m-t+i}, & i = 1, 3, \cdots, t-2, \\ \gamma_i = a'_{m-t+i} + a'_{2m-2t+i}, & i = t+1, t+3, \cdots, 2t-2, \\ \gamma_i = a'_{m+i}, & i = t, t+2, \cdots, m-2, \\ \gamma_i = a'_i + a'_{m-t+i}, & i = 2t, 2t+2, \cdots, m-1. \end{array} \tag{4}$$

The computation of intermediate values totally costs $1T_X$ in parallel. Denote by Γ the temporary value and by Θ the remainder operands of Eq. (3), namely,

$$\Theta = \sum_{i=0}^{m-1} \theta_i x^i$$

where θ_i is given by

$$\theta_i = \begin{cases} a'_i & \text{for } i = t, t+2, \cdots, 2t-2 \\ 0 & \text{otherwise .} \end{cases}$$

otherwise. Then we have $C = \Gamma + \Theta$. The forth power of $A(x)$ can be rewritten as:

$$D = A^4 \bmod f(x) = C^2 \bmod f(x) = \Gamma^2 + \Theta^2 \bmod f(x)$$

We plug in Γ and Θ in Eq. (3) and obtain the results as follows:

$$\begin{array}{ll} d_i = \gamma'_i, & i = 0, 2, 4, \cdots, t-1, \\ d_i = \gamma'_{m+i} + \gamma'_{2m-t+i} + \theta'_{m+i} + \theta'_{2m-t+i}, & i = 1, 3, \cdots, t-2, \\ d_i = \gamma'_i + \gamma'_{m-t+i} + \gamma'_{2m-2t+i} + \theta'_{m-t+i} + \theta'_{2m-2t+i}, & i = t+1, t+3, \cdots, 2t-2, \\ d_i = \gamma'_{m+i} + \theta'_{m+i}, & i = t, t+2, \cdots, m-2, \\ d_i = \gamma'_i + \gamma'_{m-t+i} + \theta'_i + \theta'_{m-t+i}, & i = 2t, 2t+2, \cdots, m-1. \end{array} \tag{5}$$

where γ'_i and θ'_i are given by

$$\gamma'_i = \begin{cases} \gamma_{\frac{i}{2}} & \text{if } i \text{ even,} \\ 0 & \text{otherwise,} \end{cases} \qquad \theta'_i = \begin{cases} \theta_{\frac{i}{2}} & \text{if } i \text{ even,} \\ 0 & \text{otherwise.} \end{cases}$$

In the above expressions, zero values of θ'_i are ignored. We can see that the third parts of Eq.(5) have five operands and the depth of the binary XOR tree is $\lceil \log_2 5 \rceil$, which will lead to $3T_X$ gates delay in parallel. Actually, note that only the θ'_i with the subscript $2(t+1), 2(t+3), \cdots, 4(t-1)$ are nonzero, if $4(t-1) < 2m-t+1$, namely, $m > \frac{5(t-1)}{2}$, then the elements $\theta'_{2m-t+1}, \theta'_{2m-t+3}, \cdots, \theta'_{2m-2}$ are all zero. At this time, Eq. (5) becomes:

$$\begin{array}{ll} d_i = \gamma'_i, & i = 0, 2, 4, \cdots, t-1, \\ d_i = \gamma'_{m+i} + \gamma'_{2m-t+i} + \theta'_{m+i}, & i = 1, 3, \cdots, t-2, \\ d_i = \gamma'_i + \gamma'_{m-t+i} + \gamma'_{2m-2t+i} + \theta'_{m-t+i}, & i = t+1, t+3, \cdots, 2t-2, \\ d_i = \gamma'_{m+i} + \theta'_{m+i}, & i = t, t+2, \cdots, m-2, \\ d_i = \gamma'_i + \gamma'_{m-t+i} + \theta'_i + \theta'_{m-t+i}, & i = 2t, 2t+2, \cdots, m-1. \end{array} \tag{6}$$

Therefore, the depth of the binary XOR tree becomes $\lceil \log_2 4 \rceil = 2$. Plus the time delay for computation of Γ, the critical path of the whole circuit takes time $3T_X$.

3.2 Complexity Analysis

Now we evaluate the cost of our approach. Note that there exists reused partial sums in the second and third parts of Eq. (3) and Eq. (4), to compute Γ and Γ^2 totally cost $\frac{m-t+1}{2}$ and $\frac{m+t-1}{2}$ XOR gates, respectively. Then it need to compute $\Gamma^2 + \Theta^2$ and get the final results. Because θ'_i only with subscript $i = 2(t+1), 2(t+3), \cdots, 4(t-1)$ are nonzero, we examined the subscripts of θ'_i in Eq. (6) and found that all the subscripts can compose two sets:

$$\{2t, 2t+2, \cdots, 2m-2\},\ \{m+1, m+3, \cdots, 2m-t-1\}.$$

Note that $m > \frac{5(t-1)}{2}$ and then $2m-2 > 4(t-1), 2m-t-1 \geqslant 4(t-1)$, it follows that the worst case is that both of the two sets have $\frac{t-1}{2}$ nonzero terms. Hence, the gates count for bit-parallel forth power here is no more than $m+t-1$ XOR gates with $3T_X$ time delay. In addition, if $m+1 > 4(t-1)$, i.e., $m > 4t-5$, the set $\{\theta'_{m+1}, \theta'_{m+3}, \cdots, \theta'_{2m-t-1}\}$ only contain zero values and the number of XOR gates required by the circuit reduced to $m + \frac{t-1}{2}$.

Consequently, we obtain the complexity of forth power as follows:

$$\#\ \text{XOR:} = \begin{cases} m + \frac{t-1}{2} & \text{if} \quad m > 4t-5, \\ m+t-1 & \text{if} \quad \frac{5(t-1)}{2} < m \leqslant 4t-5. \end{cases}$$

$$\text{Time delay:} = 3T_X$$

3.3 An Example

As a small example, we built a bit-parallel forth power in $GF(2^7)$ with generating polynomial $x^7 + x^3 + 1$. Assume that $A(x) = \sum_{i=0}^{6} a_i x^i$ is an arbitrary element in $GF(2^{10})$. Let $\sum_{i=0}^{6} c_i x^i$ denote the temporary values according to Eq. (4) and $\sum_{i=0}^{6} d_i x^i$ denote the forth power of $A(x)$. Based on previous analysis, we have:

$$\begin{aligned} c_0 &= a_0, \\ c_1 &= a_4 + a_6, \\ c_2 &= a_1, \\ c_3 &= a_5, \\ c_4 &= a_4 + a_6, \\ c_5 &= a_6 \\ c_6 &= a_3 + a_5. \end{aligned} \quad \text{and} \quad \begin{aligned} d_0 &= c_0, \\ d_1 &= c_4 + c_6 + a_2, \\ d_2 &= c_1, \\ d_3 &= c_5, \\ d_4 &= c_2 + c_4 + c_6 + a_2, \\ d_5 &= c_6, \\ d_6 &= c_3 + c_5. \end{aligned} \qquad (7)$$

The circuit is shown in Fig. 1. Note that the reused computations in above expressions can save certain gates. Then, it can be seen that seven XOR gates are used and the critical path length for forth power circuit is $3T_X$.

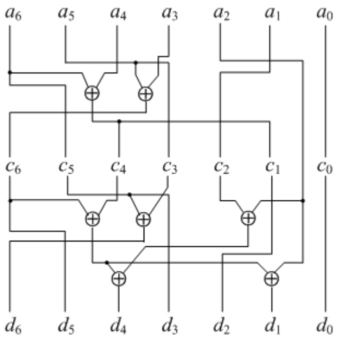

a_6 a_5 a_4 a_3 a_2 a_1 a_0
c_6 c_5 c_4 c_3 c_2 c_1 c_0
d_6 d_5 d_4 d_3 d_2 d_1 d_0

Fig. 1. Architecture for forth power with generating trinomial $x^7 + x^3 + 1$

4 Computation of Multiplicative Inverse Based on Forth Power

In this section, we will present a new version of ITA-PB based on the forth power as above. Since $2^m - 2 = 2^{m-1} + 2^{m-2} + \cdots + 2^2 + 2$ and m is odd, we have

$$2^m - 2 = (4^{\frac{m-1}{2}} + 4^{\frac{m-3}{2}} + \cdots + 4) + (4^{\frac{m-1}{2}} + 4^{\frac{m-3}{2}} + \cdots + 4)/2.$$

For any nonzero $\alpha \in GF(2^m)$, $\alpha^{-1} = \alpha^{2^m-2}$, the inversion can be written as:

$$\alpha^{-1} = \beta \cdot \beta^{\frac{1}{2}}$$

where $\beta = \alpha^{4^{\frac{m-1}{2}}+4^{\frac{m-3}{2}}+\cdots+4}$. Thus the multiplicative inverse can be computed based on forth power, squaring root and multiplication. Now we consider the computation of β. This exponentiation can be computed through repeated raising of intermediate results to the forth power and multiplications, similar with the strategy used in [6,4] to minimize the number of multiplications in $GF(2^m)$.

Theorem 1. *Let $A \in GF(2^m)$ and $0 < n < \lfloor \frac{m}{2} \rfloor$ be an integer. One can compute A^r where $r = 4^n + 4^{n-1} + \cdots + 4^2 + 4$ with no more than*

$$\begin{aligned} &\#mul := \lfloor \log_2(n) \rfloor + HW(n) - 1 \\ &\#4 - exp := n \end{aligned}$$

Operations, where $HW(\cdot)$ denotes the Hamming weight of its operand and #mul and #4-exp refer to multiplications and the forth power in the field, respectively.

Proof. First, consider the computation of A^{s_k} where $s_k = \sum_{i=1}^{2^k} 4^i$. It is easy to see that $A^{s_k} = (A^{s_{k-1}})^{4^{2^{k-1}}} A^{s_{k-1}}$. Namely,

$$\begin{aligned} A^{s_k} &= A^{\sum_{i=1}^{2^k} 4^i} = A^{\sum_{i=1}^{2^{k-1}} 4^i} \cdot A^{\sum_{i=2^{k-1}+1}^{2^k} 4^i} \\ &= A^{\sum_{i=1}^{2^{k-1}} 4^i} \cdot (A^{\sum_{i=1}^{2^{k-1}} 4^i})^{4^{2^{k-1}}} \\ &= (A^{s_{k-1}})^{4^{2^{k-1}}} \cdot A^{s_{k-1}}. \end{aligned} \tag{8}$$

Table 1. The computation of β

Power	rule	result
s_0	α^4	α^4
s_1	$(\alpha^{s_0})^4 \cdot \alpha^{s_0}$	α^{4^2+4}
s_2	$(\alpha^{s_1})^{4^2} \cdot \alpha^{s_1}$	$\alpha^{4^4+4^3+4^2+4}$
s_3	$(\alpha^{s_2})^{4^{2^2}} \cdot \alpha^{s_2}$	$\alpha^{4^8+\cdots+4^2+4}$
s_4	$(\alpha^{s_3})^{4^{2^3}} \cdot \alpha^{s_3}$	$\alpha^{4^{16}+\cdots+4^2+4}$
s_5	$(\alpha^{s_4})^{4^{2^4}} \cdot \alpha^{s_4}$	$\alpha^{4^{32}+\cdots+4^2+4}$
s_6	$(\alpha^{s_5})^{4^{2^5}} \cdot \alpha^{s_5}$	$\alpha^{4^{64}+\cdots+4^2+4}$
$4^{80}+4^{79}+\cdots+4$	$(\alpha^{s_6})^{4^{2^4}} \cdot \alpha^{s_4}$	$\gamma_1 = \alpha^{4^{80}+\cdots+4^2+4}$
$4^{88}+4^{87}+\cdots+4$	$(\gamma_1)^{4^{2^3}} \cdot \alpha^{s_3}$	$\gamma_2 = \alpha^{4^{88}+\cdots+4^2+4}$
$4^{92}+4^{91}+\cdots+4$	$(\gamma_2)^{4^{2^2}} \cdot \alpha^{s_2}$	$\gamma_3 = \alpha^{4^{92}+\cdots+4^2+4}$
$4^{94}+4^{93}+\cdots+4$	$(\gamma_3)^{4^2} \cdot \alpha^{s_1}$	$\gamma_4 = \alpha^{4^{94}+\cdots+4^2+4}$
$4^{95}+4^{92}+\cdots+4$	$(\gamma_4)^4 \cdot \alpha^{s_0}$	$\beta = \alpha^{4^{95}+\cdots+4^2+4}$

We rewrite n as its binary form $n = \sum_{i=1}^{\ell} 2^{k_i}$ with $k_1 > k_2 > \cdots > k_\ell$. Then A^r can be written as follows:

$$A^r = A^{4^n+4^{n-1}+\cdots+4^2+4} = (A^{s_{k_\ell}}) \left(\cdots (A^{s_{k_3}}) \left[(A^{s_{k_2}})(A^{s_{k_1}})^{4^{2^{k_2}}} \right]^{4^{2^{k_3}}} \cdots \right)^{4^{2^{k_\ell}}} .$$

Since $k_1 > k_i$ for $i = 2, \cdots, \ell$ then if we computed $A^{s_{k_1}}$ as above, all the other $A^{s_{k_i}}$ for $i = 2, \cdots, \ell$ will also be computed. It is easy to see that to compute $A^{s_{k_1}}$ actually needs $\lfloor \log_2 n \rfloor$ multiplications and 2^{k_1} forth powers. After we compute $A^{s_{k_1}}$, we need $\ell - 1 = HW(n) - 1$ multiplications and $2^{k_2} + \cdots + 2^{k_\ell}$ forth powers to compute Eq. (8). Adding up the partial complexity, we obtain the result in Theorem 1.

Example. Let us consider the binary field $GF(2^{191})$ generated by the irreducible trinomial $f(x) = x^{191} + x^9 + 1$. Assume that $\alpha \in GF(2^{191})$ be an arbitrary nonzero field element. We compute the multiplicative inverse $\alpha^{2^{191}-2}$. According to previous analysis, it is need to compute $\beta = \alpha^{4^{95}+4^{94}+\cdots+4}$ first and then to multiply β by $\beta^{\frac{1}{2}}$. Since $95 = 2^6 + 2^4 + 2^3 + 2^2 + 2 + 1$, based on the strategy proposed by Theorem 1, $\beta = \alpha^{s_0} \left(\alpha^{s_1} \left(\alpha^{s_2} (\alpha^{s_3} (\alpha^{s_4} (\alpha^{s_6})^{4^{2^4}})^{4^{2^3}})^{4^{2^2}} \right)^{4^2} \right)^4$. Details are illustrated in Tbl.1. The multiplicative inverse α^{-1} can be obtained as $\beta \cdot \beta^{\frac{1}{2}}$ which only needs one more multiplication and one squaring root operation.

Based on the consideration, the pseudocode of the inversion using forth power is shown in Algorithm 1.

```
Algorithm 1. Itoh-Tsujii algorithm using forth power
Require: : irreducible trinomial $f(x) = x^m + x^t + 1 (m,\ t$ odd),$\alpha \in GF(2^m)$.
Ensure: : $\alpha^{-1}$.
  Step 1. $n = \frac{m-1}{2}$
  Step 2. Exponentiation in $GF(2^m)$, yielding $\beta = \alpha^{4^n + 4^{n-1} + \cdots + 4}$.
  Step 3. Squaring root computation in $GF(2^m)$, yielding $\beta^{\frac{1}{2}}$.
  Step 4. Multiplication of $\beta$ and $\beta^{\frac{1}{2}}$.
```

The numbers of operations in $GF(2^m)$ needed by Algorithm 1 are summarized as follows:

$$\begin{aligned}&\#mul := \lfloor \log_2(\tfrac{m-1}{2}) \rfloor + HW(\tfrac{m-1}{2})\\ &\#4-exp := \tfrac{m-1}{2}\\ &\#squareroot := 1\end{aligned} \tag{9}$$

5 Discussion and Comparison

The classic ITA-PB [7] for inversion costs about $\lfloor \log_2(m-1) \rfloor + HW(m-1) - 1$ field multiplications plus $m-1$ squaring operations. If these operations are computed by single circuit respectively, we will obtain the strict bound for the complexity of ITA-PB. For instance, making use of the multiplier described in [2] and the squaring described in [7], the space complexity of ITA-PB are:

$$\begin{aligned}&\#\ \text{AND:} = m^2\\ &\#\ \text{XOR:} = m^2 + \tfrac{m-1}{2} - 1\end{aligned} \tag{10}$$

Conversely, the time complexity is given by:

$$\begin{aligned}\text{Time delay:} = \Big(\lfloor \log_2(m-1) \rfloor + HW(m-1) - 1\Big)\Big(T_A +\\ (2 + \lceil \log_2 m \rceil)T_X\Big) + 2(m-1)T_X\end{aligned} \tag{11}$$

Similarly, making use of the square root computation described in [13], we obtain the complexity of Algorithm 1 according to Eq. (9):

$$\begin{aligned}\#\ \text{AND:} &= m^2\\ \#\ \text{XOR:} &= m^2 + \tfrac{3(m-1)}{2} + t\\ \text{Time delay:} &= \Big(\lfloor \log_2(\tfrac{m-1}{2}) \rfloor + HW(\tfrac{m-1}{2})\Big)\Big(T_A +\\ &\qquad (2 + \lceil \log_2 m \rceil)T_X\Big) + \Big(3(\tfrac{m-1}{2}) + 1\Big)T_X\end{aligned} \tag{12}$$

From the Eq. (10), Eq. (11) and Eq. (12), it can be seen that the proposed algorithm requires at most $m+t$ more XOR gates but can save about $\frac{m-1}{2}T_X$ gates delay compared with the original one. Table 2 illustrates the improvement obtained with the proposed algorithm, by collecting numerical result for four

Table 2. Complexity for practical field

	trinomial	AND(#)	XOR(#)	Delay
ITA-PB	$x^{191}+x^9+1$	36,481	36,575	$12T_A+500T_X$
	$x^{217}+x^{45}+1$	47,089	47,196	$10T_A+532T_X$
	$x^{313}+x^{79}+1$	97,969	98,124	$11T_A+745T_X$
	$x^{409}+x^{87}+1$	167,281	167,484	$11T_A+937T_X$
ITA-PB (forth power)	$x^{191}+x^9+1$	36,481	36,775	$12T_A+406T_X$
	$x^{217}+x^{45}+1$	47,089	47,458	$10T_A+425T_X$
	$x^{313}+x^{79}+1$	97,969	98,516	$11T_A+590T_X$
	$x^{409}+x^{87}+1$	167,281	167,980	$11T_A+734T_X$

examples of field with cryptographic size defined by trinomial: $x^{191}+x^9+1$, $x^{217}+x^{45}+1$, $x^{313}+x^{79}+1$ and $x^{409}+x^{87}+1$.

Among the five irreducible polynomials suggested for ECC by NIST [14], there is one trinomial, namely, $x^{409}+x^{87}+1$, which is available for using the architectures proposed here. As it shows that in Tbl. 2, our algorithm roughly saves by 21% XOR gates delay compared with standard ITA-PB.

Furthermore, we have only discussed the fast forth power when the field is generated using the irreducible trinomial $f(x)=x^m+x^t+1$, with m,t being odd numbers. In fact, we have investigated the application of the same strategy for $f(x)=x^m+x^t+1$, with m being even and t being odd. When $m \geqslant \frac{5(t-1)}{2}$, the space complexity of that case is no more than $m+\frac{3t}{2}$ XOR gates and the time delay is also $3T_X$. At this time, since m is an odd number, $2^m-2=2(4^{\frac{m-2}{2}}+4^{\frac{m-4}{2}}+\cdots+4)+(4^{\frac{m-2}{2}}+4^{\frac{m-4}{2}}+\cdots+4)+2$. The main operations of inversion include forth power, multiplication and squaring. Based on the results with respect to related squaring operation in [7], we will also get the same speedup for ITA-PB in this case.

6 Conclusion

In this paper, we introduce a novel version of ITA-PB in $GF(2^m)$ which uses forth power rather than squaring as the main operation. The forth power based on ITA-PB is known to be efficient when forth power can perform faster than two serial squaring operations. In this work, by increasing the parallelism of circuit at cost of more gates, fast forth power is made possible. We have shown that the new version of ITA-PB saves about $\frac{m}{2}$ XOR gates delay with an increase of no more than $2m$ XOR gates. This proposal could provide remarkable implementation efficiency for cryptographic applications.

Acknowledgments. This work is supported by the Natural Science Foundation of China (No.60672068).

References

1. Blake, I., Seroussy, G., Smart, N.P.: Elliptic Curves in Cryptography. Cambridge University Press, Cambridge (1999)
2. Sunar, B., Koç, Ç.K.: Mastrovito Multiplier for All Trinomials. IEEE Trans. Comput. 48(5), 522–527 (1999)
3. Fournaris, A.P., Koufopavlou, O.: Applying systolic multiplication-inversion architectures based on modified extended Euclidean algorithm for $GF(2^k)$ in elliptic curve cryptography. Comput. Electr. Eng. 33(5-6), 333–348 (2007)
4. Guajardo, J., Paar, C.: Itoh-Tsujii Inversion in Standard Basis and Its Application in Cryptography. Codes. Des. Codes Cryptography 25(2), 207–216 (2002)
5. Guo, J., Wang, C.: Systolic Array Implementation of Euclid's Algorithm for Inversion and Division in $GF(2^m)$. IEEE Trans. Comput. 47(10), 1161–1167 (1998)
6. Itoh, T., Tsujii, S.: A fast algorithm for computing multiplicative inverses in $GF(2^m)$ using normal bases. Inf. Comput. 78(3), 171–177 (1988)
7. Wu, H.: Bit-Parallel Finite Field Multiplier and Squarer Using Polynomial Basis. IEEE Trans. Comput. 51(7), 750–758 (2002)
8. Yan, Z., Sarwate, D.V.: New Systolic Architectures for Inversion and Division in $GF(2^m)$. IEEE Trans. Comput. 52(11), 1514–1519 (2003)
9. Rodríguez-Henríquez, F., Morales-Luna, G., Saqib, N., Cruz-Cortés, N.: Parallel Itoh-Tsujii multiplicative inversion algorithm for a special class of trinomials. Des. Codes Cryptography 45(1), 19–37 (2007)
10. Wang, C.C., Truong, T.K., Shao, H.M., Deutsch, L.J., Omura, J.K., Reed, I.S.: VLSI Architectures for Computing Multiplications and Inverses in $GF(2^m)$. IEEE Trans. Comput. 34(8), 709–717 (1985)
11. Fan, H., Dai, Y.: Fast Bit-Parallel $GF(2^n)$ Multiplier for All Trinomials. IEEE Trans. Comput. 54(5), 485–490 (2005)
12. Sunar, B., Koç, Ç.K.: An Efficient Optimal Normal Basis Type II Multiplier. IEEE Trans. Comput. 50(1), 83–87 (2001)
13. Rodríguez-Henríquez, F., Morales-Luna, G., López, J.: Low-Complexity Bit-Parallel Square Root Computation over $GF(2^m)$ for All Trinomials. IEEE Trans. Comput. 57(4), 472–480 (2008)
14. FIPS 186-2. Digital Signature Standard (DSS). Federal Information Processing Standards Publication 186-2, National Institute of Standards and Technology (2000), http://csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf

Computational Study of Compressive Loading of Carbon Nanotubes

Yang Yang and William W. Liou

Department of Aeronautical and Mechanical Engineering
Western Michigan University
Kalamazoo, U.S
yang.yang@wmich.edu, william.liou@wmich.edu

Abstract. A reduced-order general continuum method is used to examine the mechanical behavior of single-walled carbon nanotubes (CNTs) under compressive loading and unloading conditions. Quasi-static solutions are sought where the total energy of the system is minimized with respect to the spatial degree of freedom. We provide detailed buckled configurations for four different types of CNTs and show that, among the cases studied, the armchair CNT has the strongest resistance to the compressive loading. It is also shown that the buckled CNT will significantly lose its structural strength with the zigzag lattice structure. The unloading post-buckling of CNT demonstrates that even after the occurrence of buckling the CNT can still return to its original state making its use desirable in fields such as synthetic biomaterials, electromagnetic devices, or polymer composites.

Keywords: component; carbon nanotube; finite element method; mechanical properties.

1 Introduction

Carbon nanotubes(CNTs), since first found as a form of multi-walled CNT tangles, have gained overwhelming attentions for their significant and unique properties. Many applications of CNTs have been proposed such as enhanced composite materials, efficient heat remover, precise drug delivery, etc. Extensive research efforts have been taken to understand different aspects of CNTs properties, ([1],[2],[3],[4],[5],[6],[7]) just to name a few. However there are limited research activities recorded on the subject of fatigue analysis of CNTs under cyclic loading conditions. The knowledge of CNT's fatigue characteristics becomes indispensable for its application in a variety of nanoscale materials and devices such as CNT actuator and CNT-based electromechanical switch, where nanotubes may be subject to 10^6 cycles during their lifetime[8].

Mechanical behaviors of a forest of multi-walled CNTs subject to the nanoindenter loading was reported by [9]. The experiment captured shell buckling mode of the CNTs and produced a series of hysteresis loops from loading-unloading processes. A distinct drop in slope of the loop is evident because of the onset

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 25–43, 2010.

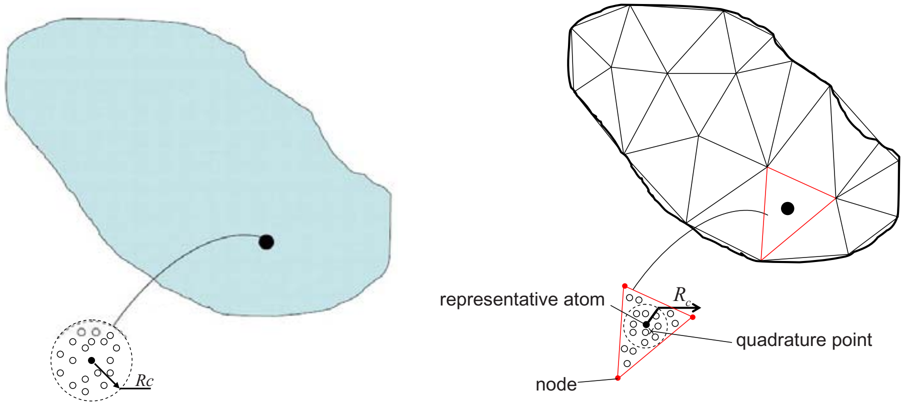
of the buckling. [10] investigated the cyclic deformation behavior of the single-walled CNTs/epoxy composites. It was found that the size of the hysteresis loop between the loading and unloading cycles decreased with the increase of the deformation cycles. [11] used the molecular dynamics (MD) method to determine the hysteresis loops of CNTs of different sizes under cyclic tensile and compressive loading conditions. It was shown that the CNT exhibited extreme cyclic loading resistance with yielding strain and strength becoming constant after a number of loading cycles.

In this paper, a quasi-static reduced-order general continuum method [12] is used to calculate the yielding stress and strain of CNT. The method uses the widely-accepted atomic potential, the reactive empirical bond order potential (REBO) for hydrocarbon molecules [13] as the base model to calculate the interatomic energy among carbon atoms in CNT. The REBO potential includes the bond order terms which implicitly describe the angular dependence of interatomic forces so that the computationally expensive operations of registering and tracking many atoms for multibody interactions are avoided. The quasi-static state of CNT is determined by seeking the minimum configuration of the energy potential. The compressive deformation of CNT is simulated by prescribing the displacement boundary conditions to each node on both ends of the CNT or a displacement-controlled method.

2 Numerical Method

The reduced-order general continuum method combines the $\mathbf{C}^1$ subdivision finite element method [14] and the quasi-continuum scheme [15] that links the crystal lattice deformation on atomic scale to the nonlinear solid mechanics deformation on macroscopic scale. With fourth-order shape functions, the subdivision finite element method interpolates fields not only by nodes from a local triangular element but also by nodes from neighboring elements, which guarantees a higher-degree of continuity in interpolation than lower order methods do and avoids shear locking problems that usually take place in conventional high-order methods. The method has been successfully applied in solving many thin-shell problems, such as [16], [17], [18].

The quasi-continuum scheme stipulates that the relation between the quantities of the two different scales can only be established via the deformation gradient $\mathbb{F}$. For any macroscopic point on a continuum body, it is represented on atomic scale by a crystallite of radius R_c with infinite number of atoms embedded as shown in Fig. 1 (a). The crystalline lattice deforms according to the local continuum displacements. And the energy of the crystallite is determined directly from an appropriate atomic model so that the key properties of the crystal, such as atomic symmetry, are preserved. In a finite element method, the continuum solid is discretized into many small elements, as shown in Fig. 1 (b). The displacements in an element are interpolated from the values on the corresponding nodal



(a) Every point in the continuum body is described by a representative atom embedded in a crystallite of radius R_c.

(b) Finite element discretizing the continuum body.

Fig. 1. Schematics showing the linkage between atomic lattice and macroscopic finite element method

points, and they become the only major unknowns to the numerical solution. The Cauchy-Born rule is used to relate the macroscopic deformation of crystals to the changes in the lattice vectors. In the current research, we consider all deformations in CNT to be homogeneous meaning that the CNT deforms strictly according to a universal deformation gradient. The choice of representative crystallite is immaterial to the calculation. We can always choose the one that is most convenient to the finite element formulation or the representative cell. The structure of the representative cell is shown in Fig. 2 where $a_{1\ldots3}$ and $\theta_{1\ldots3}$ are the three unique bond lengths and angles for this cell upon which the interatomic energy depends. The specialized form of the REBO potential function is shown as follows

$$E = \sum_{i=1}^{3} \left[V_R(a_i) - B_i(a_j, a_k, \theta_j, \theta_k) V_A(a_i) \right] \tag{1}$$

where $V_R(a_i)$ is the repulsive pair term and is given as

$$V_R(a_i) = f_i(a_i) \frac{D_{CC}^{(e)}}{S_{CC} - 1} e^{-\sqrt{2S_{CC}}\beta_{CC}(a_i - R_{CC}^{(e)})}; \tag{2}$$

$V_A(a_i)$ the attractive pair term and is given as

$$V_A(a_i) = f_i(a_i) \frac{D_{CC}^{(e)} S_{CC}}{S_{CC} - 1} e^{-\sqrt{2/S_{CC}}\beta_{CC}(a_i - R_{CC}^{(e)})}; \tag{3}$$

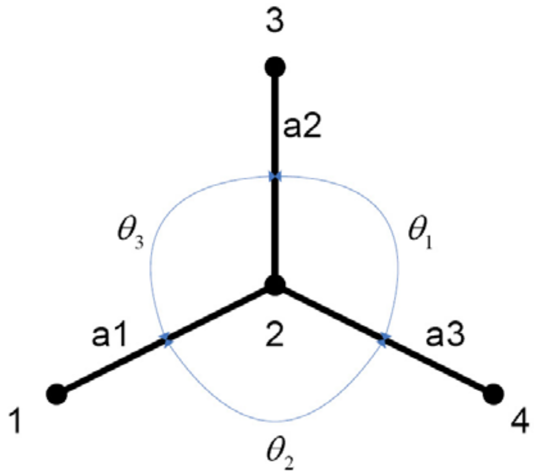


Fig. 2. Local structure of a CNT

and B_i is the bond order term which is given as

$$B_i = \left[1 + G_C(\theta_k) f_j(a_j) + G_C(\theta_j) f_k(a_k)\right]^{-\delta} . \tag{4}$$

The detailed expressions for every term in the formalism can be found in [13]. The method of calculating the bond lengths and angles is based on the approximate exponential map of carbon sheet proposed by [12]. A brief account of the method is given below for completeness.

As discussed in the previous paragraph, Cauchy-Born rule relates the atomic lattice vector to the macroscopic deformation through a simple relation as shown in Fig. 3. This is appropriate for bulk materials, such as the cubic lattice. However when it comes to the graphene sheet studied here, which is a two-dimensional manifold embedded in a three-dimensional space, the Cauchy-Born rule can not be used directly. By referring to Fig. 4, the deformation gradient $\mathbb{F}$ only maps the infinitesimal line segment between tangent spaces. The lattice vector is in

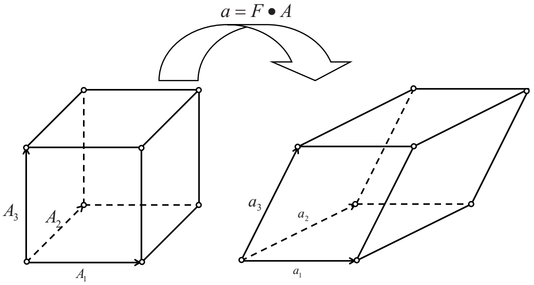


Fig. 3. Cauchy-Born rule applied to a cubic lattice

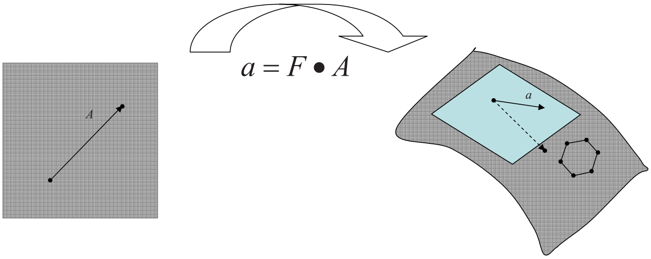


Fig. 4. Direct use of Cauchy-Born rule leads to inaccurate results

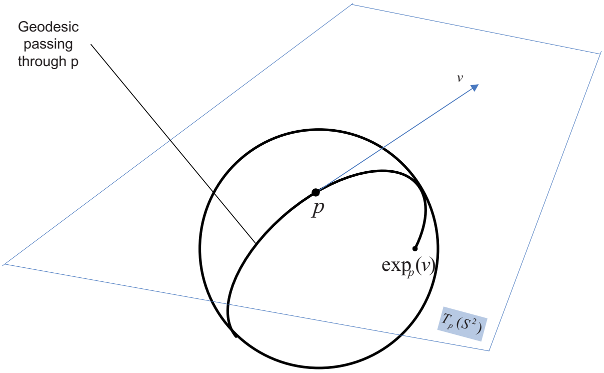


Fig. 5. Schematics showing exponential map

fact the chord of the carbon sheet which is not correctly captured by this simple relation. The effect of exponential map is to pull away a vector **v** in the tangent space to a chord **A** of the geodesic circle, as shown in Fig. 5. At point P on the surface of the globe, there are a tangent vector **v** and the geodesic circle passing through it. The exponential map $\mathbf{A} = \exp_P(\mathbf{v})$ lays off a length equal to $\|\mathbf{v}\|$ along the geodesic passing P in the direction of **v** to form a new vector **A**, that is the chord of the geodesic circle.

Fig. 6 shows the transformation for the current case. **A** represents the bond vector connecting atoms X and Y on the carbon sheet, and **W** the tangent bond vector for the un-deformed configuration. Similarly, **a** represents the bond vector connecting atoms x and y on the carbon sheet, and **w** the tangent bond vector for the deformed configuration. Since the deformation gradient $\mathbb{F}$ only maps entities in between the tangent spaces, i.e. $\mathbf{w} = \mathbb{F}\mathbf{W}$, it is necessary to first use the inverse exponential map $\exp_X^{-1}$ to map the chord vector **A** into the tangent space then after the deformation gradient operation use the exponential map $\exp_x$ again to obtain the desired chord vector **a**. The above concept can be expressed in the following composite mapping relation

$$\mathbf{a} = \exp_{\phi(X)} \circ \mathbb{F}(\mathbf{X}) \circ \exp_X^{-1}(\mathbf{A}). \tag{5}$$

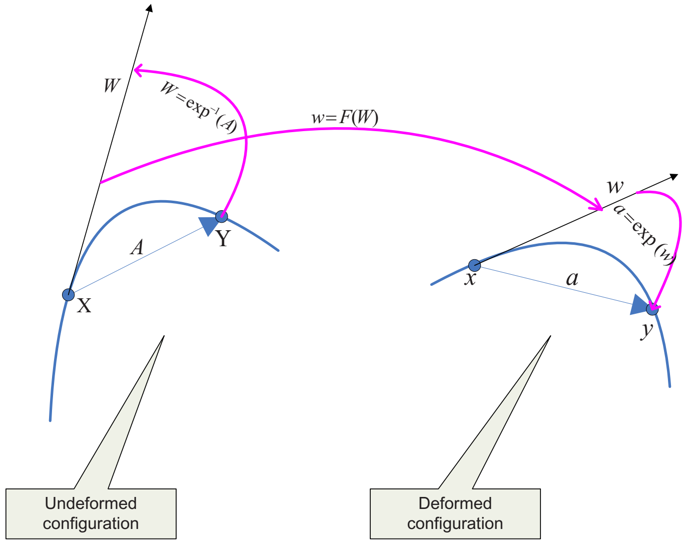


Fig. 6. Schematics showing exponential map between two configurations

The order of composite operation is from the right most towards the left. Since carbon nanotubes are made, virtually by rolling up an originally flat graphene sheet into the cylindrical shape, it is reasonable and convenient to consider the flat carbon sheet as the reference configuration. Eqn. 5 can be simplified as

$$\mathbf{a} = \exp_{\phi(X)} \circ \mathbb{F}(\mathbf{X}) \circ (\mathbf{A}), \tag{6}$$

since the chord vector and the tangent vector are coincident on a flat sheet. The evaluation of exponential map needs the knowledge of geodesic circles which involves solving a system of two nonlinear ODE's. And the coefficients of the equations are Christoffel symbols. Analytical approaches [19] have not been successful in finding their expressions. The approximate exponential map method decouples the general deformation into two principal directions $\mathbf{V}_1$ and $\mathbf{V}_2$. And the method assumes that the curved surface can be represented locally in $\mathbf{V}_{1,2}$ by a cylinder with the radii of $1/k_{1,2}$ or the inverse of the principal curvature. The exponential map for a cylindrical surface has a closed-form solution which can be used to find the approximate deformed bond vector. The detailed derivation is referred to [20]. Only the final expressions for bond lengths and angles are given here.

$$\mathbf{a} = \begin{bmatrix} a^1 \\ a^2 \\ a^3 \end{bmatrix} = \begin{bmatrix} w^1 \frac{sin(k_1 w^1)}{k_1 w^1} \\ w^2 \frac{sin(k_2 w^2)}{k_2 w^2} \\ \frac{k_1 (w^1)^2}{2} \frac{sin^2\left(\frac{k_1 w^1}{2}\right)}{\left(\frac{k_1 w^1}{2}\right)^2} + \frac{k_2 (w^2)^2}{2} \frac{sin^2\left(\frac{k_2 w^2}{2}\right)}{\left(\frac{k_2 w^2}{2}\right)^2} \end{bmatrix} \tag{7}$$

the terms $w^{1,2}$ in Eqn. 7 are the components of the tangent deformed bond vector $\mathbf{w}$ that are resolved in the two principal directions $\mathbf{v}_{1,2}$ and they are given as

$$\mathbf{w} = \begin{bmatrix} w^1 \\ w^2 \end{bmatrix} = \begin{bmatrix} C_{AB}A^A(V_1)^B \\ C_{AB}A^A(V_2)^B \end{bmatrix}, \tag{8}$$

where C_{AB} are the components of Green strain tensor. The length of the deformed bond vector is simply

$$a = \|\mathbf{a}\|, \tag{9}$$

and the angle between any two bond vectors $\mathbf{a}$ and $\mathbf{b}$ can be found by

$$cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}. \tag{10}$$

The interatomic energy density W is

$$W = \frac{E}{S_0}, \tag{11}$$

where S_0 is the area of the representative cell and is given as

$$S_0 = \frac{3\sqrt{3}}{2}\|A_{0i}\|^2, \tag{12}$$

where $\mathbf{A}_{i0}$ is the undeformed carbon-carbon bond length. The energy due to REBO potential or the short-range interaction can be found by integrating the energy density over the CNT surface as follows

$$E_{\text{REBO}} = \int_{\Omega_0} W d\Omega_0, \tag{13}$$

where Ω_0 represents the reference configuration.

The long-range van der Waals energy E_{LJ} in CNT is modelled by the modified Lennard-Jones potential. If two representative cells are considered, labeled as I and J, as shown in Fig. 7. There are two carbon atoms in each cell, then the Lennard Jones potential energy can be calculated as follows

$$E_{\text{LJ}} = \sum_{I=1}^{4} \sum_{J>I, J \notin B_I}^{4} \mathcal{V}(\|\mathbf{r}_{IJ}\|), \tag{14}$$

where $\mathbf{r}_{IJ}$ is the vector that connects two nonbonded atoms located within different representative cells; B_I is a group of atoms that are bonded to atom I. The computation is looped over two nonbonded atoms. However Eqn. 14 can only be carried out over discrete individual atoms. For the purpose of finding the total Lennard-Jones energy about the surface of carbon nanotubes, a continuum version of Eqn. 14 is needed. Based on the assumption that he carbon atoms are distributed homogeneously on the CNT surface and undergoes a homogeneous

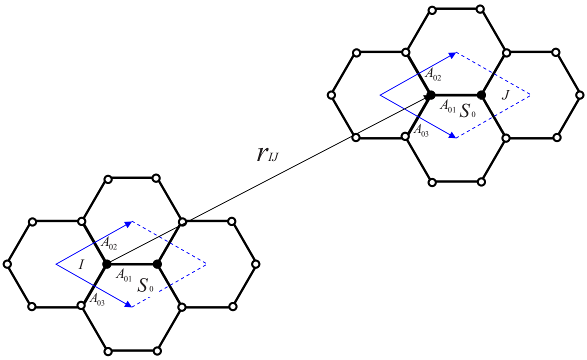


Fig. 7. Schematics showing Lennard Jones potential calculation over the representative cells

deformation. The mean surface density for carbon atoms, $\mathcal{M}$ is $\mathcal{M} = 2/S_0$. The practical Lennard Jone potential formula is

$$E_{LJ} = \frac{4}{S_0^2} \int_{\Omega_{0I}} \int_{\Omega_{0J}-B_I} \mathcal{V}(\|\mathbf{r}_{IJ}(\mathbf{X}_I, \mathbf{X}_J)\|) d\Omega_{0I} d\Omega_{0J}, \tag{15}$$

where $\mathbf{X}_I$ and $\mathbf{X}_J$ are the position vectors for particles (not necessarily coincident with carbon atoms) on the continuum surface defined by the representative cell I or J. As a result the total van der Waals potential energy between two interacting surfaces can be found through a double integration. The exact van der Waals potential expression takes the following format

$$\mathcal{V} = \begin{cases} 0.0 & r_{ij} \leqslant r_{\text{small}} \\ c_{3,k}(r_{ij} - r_k)^3 + c_{2,k}(r_{ij} - rk)^2 & \\ \quad + c_{1,k}(r_{ij} - r_k) + c_{0,k} & r_{\text{small}} < r_{ij} < r_{\text{medium}} \\ 4\epsilon \left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6} \right] & r_{\text{medium}} \leqslant r_{ij} \leqslant r_{\text{big}} \end{cases} \tag{16}$$

The parameters for Lennard-Jones potential are taken from [21] and listed in Table 1. Parameters $c_{n,k}$ and r_k are the coefficients of the cubic spline interpolation function, which can be found in [21]. Eqn. 16 includes a third order polynomial as the switching function so that both the function and the gradient evaluations of the Lennard-Jones potential are well-defined, which reduces overshooting near the lower bond of the cutoff radius.

Considering an external energy applied to CNT, the total energy of the system is

$$E_{\text{total}} = E_{\text{REBO}} + E_{\text{LJ}} - E_{\text{external}}. \tag{17}$$

The equilibrium or the stable state of a CNT is found by solving a minimum value problem with E_{total} being treated as the objective function. The standard optimization subroutine BFGS [22], [23] is used for this purpose. A displacement-controlled method is used to simulate the compression test with both ends being

Table 1. Parameters for carbon-carbon Lennard-Jones potential

Parameter	Value
ϵ	4.2038×10^{-3} eV
σ	$3.37\dot{A}$
r_{small}	$2.28\dot{A}$
r_{medium}	$3.40\dot{A}$
r_{big}	$6.8\dot{A}$

displaced at half of the prescribed value. For every new displaced configuration of CNT, the total energy E_{total} is minimized so that the equilibrium state can be achieved for this specific condition.

3 Simulation Results

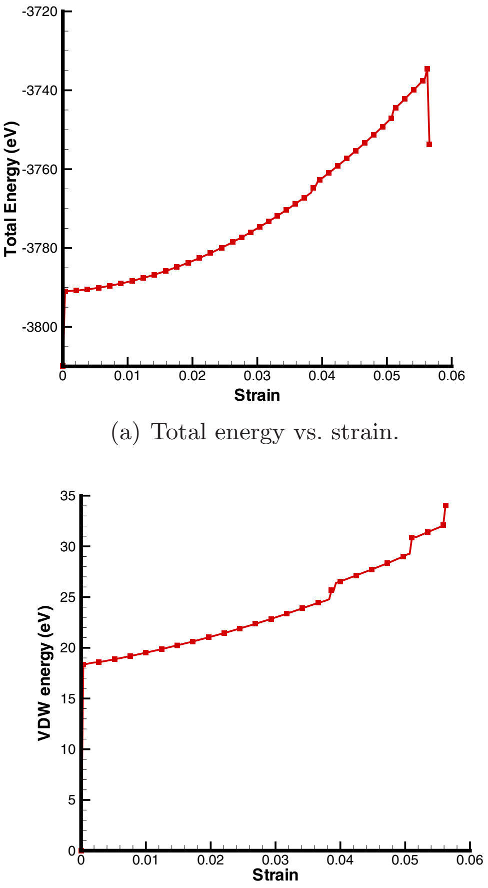
With the numerical method introduced in the previous section, we simulated the buckling phenomenon occurred in single-walled CNTs under compression loading. Before any simulation, the CNT will first be relaxed to an equilibrium state without any external load being applied. The equilibrium state corresponds to the lowest energy achievable by the CNT under such a configuration. This initial equilibration process will eliminate any residual stress gained during the wrapping process (from a flat graphene sheet to the tube). Four different configurations of CNT are considered as shown in Table 2, which cover the range of possible characteristic configurations of carbon lattice structure in relation to the applied loading with the chiral angle varying from 0° to 30°.

Carbon nanotubes under axial compression load will ultimately experience some form of morphological change because of the bucklings occurring at either local or global level. Buckling in CNT results in a sudden drop of the total energy and is followed by a weakened structural strength of CNT. It is of crucial importance to have a better understanding of the phenomenon in order to predict the structural stability of CNT.

In this section we present a series of numerical simulation results for Case 1 through Case 4. In the simulation, a CNT is subjected to an axial compression

Table 2. CNT Cases Studied

Case	Configuration Number	Length	Chiral Angle
1	$(14, 0)$ *zigzag*	40.3Å	0°
2	$(12, 3)$	40.5Å	10.9°
3	$(10, 5)$	36.8Å	19.1°
4	$(8, 8)$ *armchair*	40.9Å	30°



(a) Total energy vs. strain.

(b) van derWaals energy vs. strain.

Fig. 8. Energy variation with respect to strain during loading for Case 1

load. The load is realized numerically by prescribing the nodes on each end of the tube with an equal and opposite axial displacement increment. To establish stable gripping in numerical simulations, two extra layers of mesh points next to the loading end are also prescribed with the same displacement conditions. For the formulation used in this study the second Piola-Kirchhoff (PK2) stress tensor is the primary stress tensor. The PK2 stress tensor describes the state of stress in the undeformed configuration. The engineering strain is used as the strain measure. CNTs of all four cases are compressed axially at the same step size. The step size of the incremental compression is selected such that the onset of buckling is captured, while maintaining the stability of the energy minimization calculations. The total energy E_{total} of the new configuration is minimized with respect to the degrees of freedom for each compression step. E_{total} generally serves as a good monitor to detect the occurrence of buckling

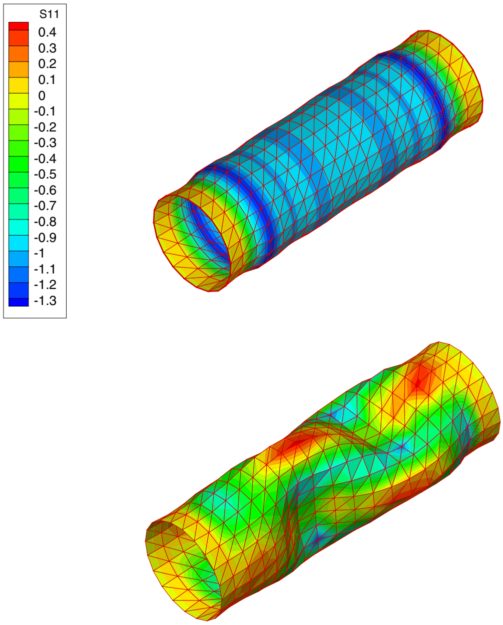


Fig. 9. Buckling patterns of CNT with different configurations, case 1

Table 3. Buckling Strain for Different CNTs

Case	Configuration Number	Chiral Angle	Strain at Buckling
1	$(14, 0)$ *zigzag*	0°	0.056
2	$(12, 3)$	10.9°	0.059
3	$(10, 5)$	19.1°	0.069
4	$(8, 8)$ *armchair*	30°	0.08

as shown in Fig. 8 (a). Before buckling, the total energy in Case 1 grows almost quadratically with strain. When the buckling happens, the CNT releases about 18.35 eV of energy. The long-distance interaction is accounted for in the form of van der Waals (vdW) force. The energy from vdW force (Fig. 8 (b)) is a small fraction of the total strain energy as demonstrated in Fig. 8. The three steps seen in the vdW energy curve corresponds buckling events taking place during the compression process. The first two are caused by the small local buckling and the third one by the global buckling. The vdW force plays an important role near buckling because the energy release of the total energy is on the same order of magnitude as the vdW energy. The numerical simulations will result in unphysical results without considering the vdW energy.

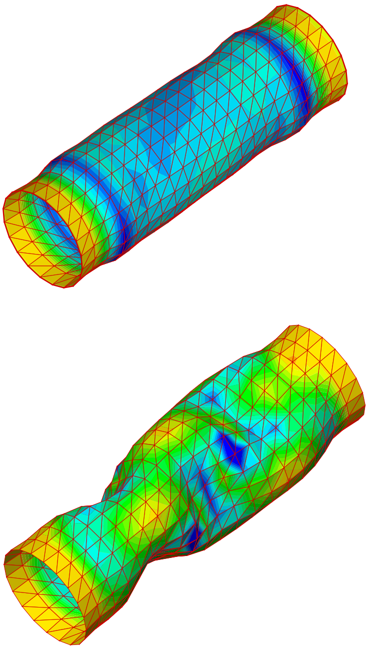


Fig. 10. Buckling patterns of CNT with different configurations, case 2

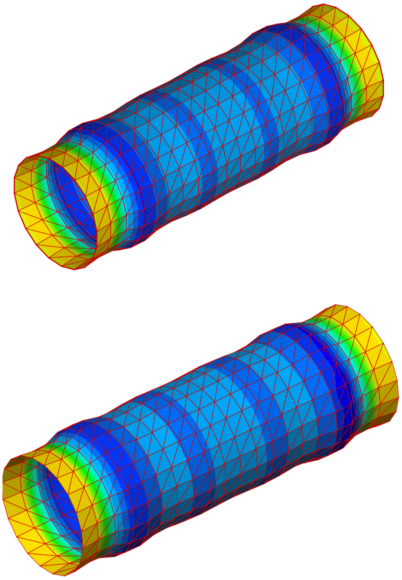


Fig. 11. Buckling patterns of CNT with different configurations, case 3

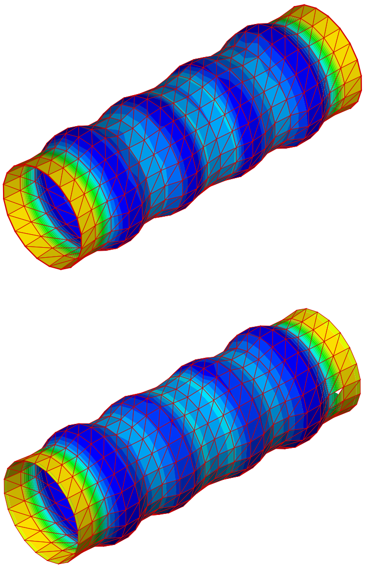
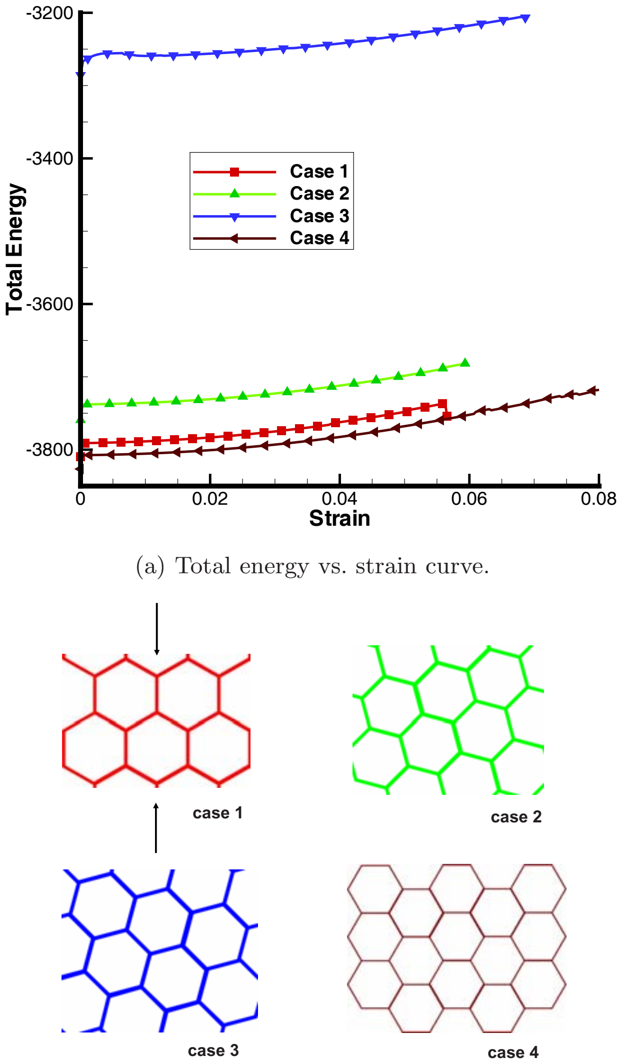


Fig. 12. Buckling patterns of CNT with different configurations, case 4

Fig. 9 through 12 show that the morphological change from the incipient buckling state (the upper one in each figure) to the buckled state (the lower one in each figure) diminishes with the increasing chiral angle, which indicates that the energy drop after buckling will also diminish.

The color contours represent the distribution of the PK2 normal stresses, S_{11}. Fig. 10 through 12 bear the same contour legend as Fig. 9. The negative stress value is considered to be of compressive sense while the positive means tension. The stress unit of force per unit length was used in that no tubule thickness enters into the formulation in this study. The two dimensional manifold nature of carbon sheet has been taken into account by the approximate exponential map method as described in Section II.

For Case 1 where the chiral angle is 0°, the buckled configuration distinguishes itself from the rest by a set of two symmetric fins formed perpendicular to each other about the tubule axis. Since three layers of mesh lines from the ends of CNT were constrained from having any freedom of deformation, the stress in these griping ends is uniformly zero over the course of compression for each case. Case 2 CNT has a chiral angle of 10.9° and presents two symmetric fins in the buckled configuration similar to Case 1 but not perpendicularly aligned with each other any more. With the increase of the chiral angle, the fin patterns disappeared in Cases 3 and 4. Instead, the buckled configurations present axisymmetric buckling pattern, as demonstrated in Fig. 11 and 12. The change in the buckling patterns among the different CNTs are attributed to the decrease of the axial strength



(a) Total energy vs. strain curve.

(b) Different orientation of carbon lattice under different chiral angles.

Fig. 13. Relations between CNT lattice structure and total energy variation

of CNT with the increase of chiral angle, which will be further discussed in the text that follows.

Fig. 13 (a) shows the changes of the total energy in a CNT with respect to loading strain for the four cases. The loading process is stopped when a buckling event is detected. Fig. 13 (b) shows the characteristic atomic lattice structures of the four types of CNTs studied here. The lattice structure demonstrates how

Table 4. Values of a

Case	Configuration Number	Chiral Angle	a
1	$(14, 0)$ *zigzag*	0°	19.04
2	$(12, 3)$	10.9°	18.83
3	$(10, 5)$	19.1°	18.26
4	$(8, 8)$ *armchair*	30°	17.47

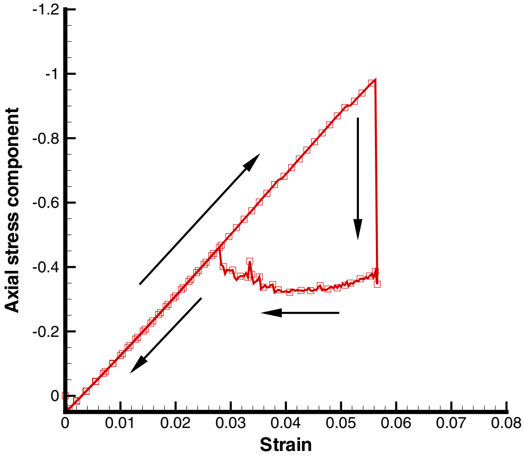


Fig. 14. Stress-strain curve for CNTs in loading-unloading conditions, case 1

a carbon bond deviates from the CNT's axial direction (aligned with the solid black arrows in the figure) in the four cases studied. The compression loading is applied in the direction of the solid black arrows.

It can be seen that the strain at which the CNT buckles changes with the chiral angle. As shown in Table 3, the buckling strain increases with the increase of the chiral angle of CNT. The cold mechanism proposed by [24] can be used to explain this observation. The simulations performed in this study are for absolute zero temperature and no temperature effects have been considered. At low temperature, thermal fluctuations are insignificant and the yield event is purely mechanical. Based on Cauchy-Born rule of homogeneous deformation of CNTs, the gross elastic strain can be associated with the individual bond elongation as pointed out by [24] and the yielding strain increases with the chiral angle. In other words, the armchair CNT has the strongest resistance to the compressive loading while the zigzag CNT has the least.

The behavior of the CNT during post-buckled unloading process is also studied. Once a buckling event is detected, the strain-controlled method is used to unloaded the CNT from compression by axially moving both ends away from each other at the same step size as used in the loading process. The stress-strain curve is plotted

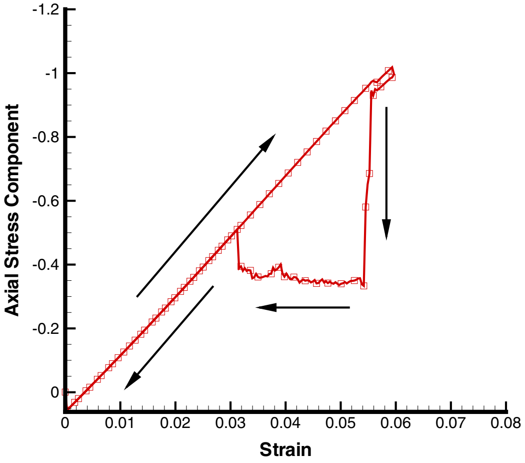


Fig. 15. Stress-strain curve for CNTs in loading-unloading conditions, case 2

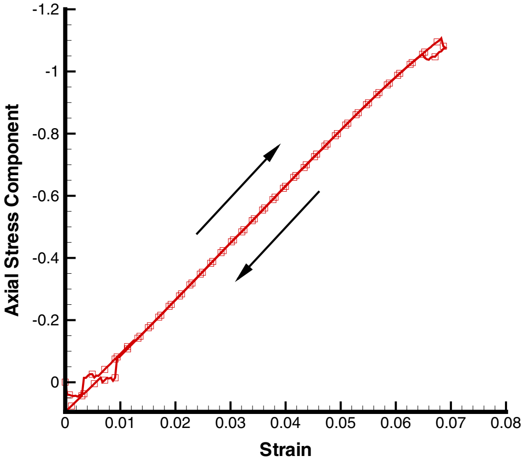


Fig. 16. Stress-strain curve for CNTs in loading-unloading conditions, case 3

for each case in Figs. 14 through 17. The stress values are calculated by averaging the corresponding axial components of PK2 stress tensor at the nodes along the circumference half way between the end planes of the CNT. The loading and unloading directions are indicated by solid black arrows in each figure. For each of the cases simulated, after an initial transient with a small stress increase, there is a region of the stress-strain curve that is essentially linear, i.e. $S_{11} = a\epsilon$. The value of the slope a for all four cases are shown in Table 4. The slope a decreases with the increasing of the chiral angles. Nonlinearity appears near the end of the linear portion of the stress-strain curve which is attributed by the fact that minor local

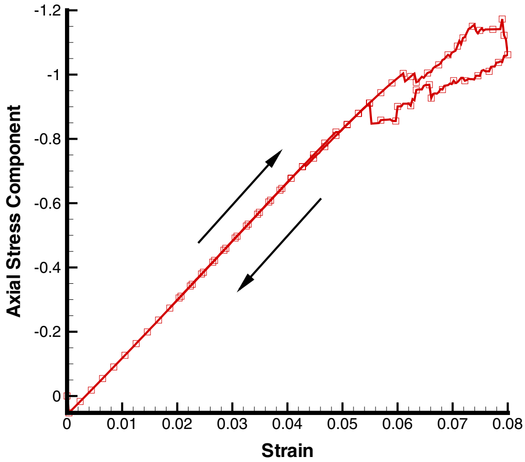


Fig. 17. Stress-strain curve for CNTs in loading-unloading conditions, case 4

bucklings have occurred especially for Cases 2 through 4 where the appearance of wavy structures precedes the global buckling. For Case 1, the buckling happens in such a clean-cut fashion that little nonlinear behavior is observed before the global buckling occurs. The structural strength of CNTs of Cases 1 and 2 weakens significantly after buckling and only mild softening occurred to Cases 3 and 4. During unloading, it is found that after exhibiting plastic-like behavior initially(the flat portion of the stress-strain curve in Cases 1 and 2), the CNT will return to its original loading path without residual strains after this loading-unloading cycle is completed. The strain energy stored during the loading process is greater than the strain energy dissipated during the unloading process. These strain energy losses are higher for Cases 1 and 2 when compared with those of Cases 3 and 4. The phenomena could be related to the thermal energy transfer between the tube and its environment which is not modeled in the simulation performed here.

4 Conclusion

We have used the reduced order general continuum method to examine the behaviors of CNTs under compressive loading-unloading conditions. We show that, with different chiral angles, different buckled configurations will be assumed by CNTs. Zigzag CNT has the most apparent buckling process than the other simulated CNT configurations with larger chiral angles. The buckling strain increases with the increasing chiral angle. Armchair CNT has the strongest resistance to compressive loading. The loading-unloading simulations show that the strength of CNT is reduced due to the compressive loading after buckling, especially for Cases 1 and 2. However, during unloading, the CNTs simulated return to its original state, demonstrating a unique mechanical behavior of single-walled CNT in response to compressive loadings.

References

1. Iijima, S.: Helical microtubules of graphitic carbon. Nature 354, 56–58 (1991)
2. Iijima, S., Maiti, C.J.: Structural flexibility of carbon nanotubes. Journal of Chemical Physics 104, 2089–2092 (1996)
3. Qian, D., Wagner, G.J., Liu, W.K., Yu, M., Ruoff, R.S.: Mechanics of carbon nanotubes. Appl. Mech. Rev. 55(6), 495–532 (2002)
4. Wei, B.Q., Vajtai, R., Jung, Y., Ward, J., Zhang, R., Ramanath, G., Ajayan, P.M.: Organized assembly of carbon nanotubes. Nature 416, 495–496 (2002)
5. Harris, P.J.F.: Carbon nanotubes and related structures. Cambridge University Press, Cambridge (1999)
6. Reich, S., Thomsen, C., Maultzsch, J.: Carbon nanotubes, basic concepts and physical properties. WILEY-VCH, Chichester (2004)
7. Yakobson, B.I., Brabec, C.J., Bernholc, J.: Nanomechanics of carbon tubes: instabilities beyond linear response. Physical Review Letters 76(14), 2511–2514 (1996)
8. Li, C., Thostenson, E.T., Chou, T.-W.: Sensors and actuators based on carbon nanotubes and their composites: A review. Composites Science and Technology 68, 1227–1249 (2008)
9. Waters, J.F., Guduru, P.R., Jouzi, M., Xu, J.M., Hanlon, T., Suresh, S.: Shell buckling of individual multiwalled carbon nanotubes using nanoindentation. Applied Physica Letters 87, 103–109 (2005)
10. Kao, C.C., Young, R.J.: Modeling the stress transfer between carbon nanotubes and a polymer matrix during cyclic deformation. In: Pyrz, R., Rauche, J.C. (eds.) IUTAM Symposium on Modelling Nanomaterials and Nanosystems, pp. 211–220 (2009)
11. Wang, J., Gutierrez, M.S.: Molecular simulations of cyclic loading behavior of carbon nanotubes using the atomistic finite element method. Journal of Nanomaterials (2009)
12. Arroyo, M., Belytschko, T.: An atomistic-based finite deformation membrane for single layer crystalline films. Journal of Mechanics and Physics of Solids 50, 1941–1977 (2002)
13. Brenner, D.W.: Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films. Physical Review B 42(15), 9458–9471 (1990)
14. Cirak, F., Ortiz, M., Schröder, P.: Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. International Journal for Numerical Methods in Engineering 47, 2039–2072 (2000)
15. Tadmor, E.B., Ortiz, M., Phillips, R.: Quasicontinuum analysis of defects in solids. Philosophical Magazine 73(6), 1529–1563 (1996)
16. Cirak, F., Ortiz, M., Pandolfi, A.: A cohesive approach to thin-shell fracture and fragmentation. Comput. Methods Appl. Mech. Engrg. 194, 2604–2618 (2005)
17. Deiterding, R., Cirak, F., Mauch, S.P., Meiron, D.I.: A virtual test facility for simulating detonation-induced fracture of thin flexible shells. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3992, pp. 122–130. Springer, Heidelberg (2006)
18. Cirak, F., Deiterding, R., Mauch, S.P.: Large-scale fluid-structure interaction simulation of viscoplastic and fracturing thin-shells subjected to shocks and detonations. Computers and Structures 85, 11–14 (2007)
19. do Carmo, M.P.: Differential geometry of curves and surfaces. Prentice-Hall, Inc., Englewood Cliffs (1976)

20. Yang, Y.: Atomistic-based finite element simulation of carbon nanotubes. Ph.D. dissertation, Western Michigan University, Kalamazoo, MI 49008 (December 2008)
21. Sinnott, S.B., Shenderova, O.A., White, C.T., Brenner, D.W.: Mechanical properties of nanotubule fibers and composites determined from theoretical calculations and simulations. Carbon 36(1-2), 1–9 (1998)
22. Byrd, R.H., Lu, P., Noceda, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM, J. Scientific Computing 16(5), 1190–1208 (1995)
23. Zhu, C., Byrd, R.H., Lu, P., Nocedal, J.: L-bfgs-b: a limited memory fortran code for solving bound constrained optimization problems. EECS Department, Northwestern University, Tech. Rep. NAM-11 (1994)
24. Dumitrica, T., Hua, M., Yakobson, B.I.: Symmetry-, time-, and temperature-dependent strength of carbon nanotubes. PNAS 103(16), 6105–6109 (2006)

plrint5d: A Five-Dimensional Automatic Cubature Routine Designed for a Multi-core Platform

Tiancheng Li and Ian Robinson

La Trobe University, Melbourne, Australia
{t.li, i.robinson}@latrobe.edu.au

Abstract. *plrint5d* is an automatic cubature routine for evaluating five-dimensional integrals over a range of domains, including infinite domains. The routine is written in C++ and has been constructed by interfacing a three-dimensional routine with a two-dimensional routine. It incorporates an adaptive error control mechanism for monitoring the tolerance parameter used in calls to the inner routine as well as multi-threading to maximize performance on modern multi-core platforms. Numerical results are presented that demonstrate the applicability of the routine across a wide range of integrand types and the effectiveness of the multi-threading strategy in achieving excellent speed-up.

Keywords: Multi-dimensional integration, automatic cubature routine, lattice augmentation sequence, multi-threading.

1 Introduction

r2d2lri [1] and *elrint3d* [2] are automatic cubature routines for evaluating two-dimensional and three-dimensional integrals, respectively. Each routine is based on non-adaptive use of a lattice augmentation sequence, a sequence of embedded lattices resulting from $(2^k)^s$-copying, $k = 1, 2, \ldots$, where s is the number of dimensions. Numerical results show that the two routines are effective for a wide range of integrand types and integration domains, including infinite domains. A particular feature of each routine is that, following application of a periodising transformation, the rapid convergence of the sequences used renders the routines very fast for even the most difficult integrals.

The methodology used in *r2d2lri* and *elrint3d* can be extended to higher dimensions in a very natural way. However, both the search for good augmentation sequences in higher dimensions and the exponential increase in the number of lattice points resulting from $(2^k)^s$-copying in higher dimensions renders the approach prohibitively expensive in dimensions greater than 4. We take a different approach in this paper by demonstrating how the two low-dimension routines can be used in combination to evaluate five-dimensional integrals. The approach adopted lends itself to use of multi-threading in a multi-core environment in a straightforward way, resulting in a very efficient routine when high accuracies are required.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 44–59, 2010.

The routine is called *plrint5d* (product lattice rule integration, 5 dimensions). Although it is not strictly based on a single product rule, it may be seen as using a product of the two routines *elrint3d* and *r2d2lri*. It is designed to evaluate (iterated) integrals of the very general form

$$If = \int_a^b \int_{c(x)}^{d(x)} \int_{g(x,y)}^{h(x,y)} \int_{p(x,y,z)}^{q(x,y,z)} \int_{r(x,y,z,u)}^{s(x,y,z,u)} f(x,y,z,u,v)\ dvdudzdydx,$$

where a and b are constants, possibly infinite, and the values of any of the functions c, d, g, h, p, q, r and s may be (and commonly are) constant, again possibly infinite.

We provide a brief description of the main features of *r2d2lri* and *elrint3d* in the next section. (The reader can find further information on lattice augmentation sequences and on how the sequences are used in these two routines in [1] and [2].) In the succeeding sections, we give an overview of the design of *plrint5d*, describe the multi-threading and error control strategies and provide some examples of the use and performance of the routine.

2 The Component Routines

The algorithm *r2d2lri* is written in C++. It is based on a lattice augmentation sequence that uses the 89-point Fibonacci lattice as seed lattice (see [3]) and a sixth-order trigonometric periodising transformation due to Sidi [4]. The routine is effective for all but the most difficult integrals and is extremely fast. It is well suited to the evaluation of integrals with either vertex or line singularities on the boundary of the integration domain. It is not suitable for integrals containing a discontinuity in either the integrand or its derivative, nor for integrals containing an internal singularity. In practice, one would normally subdivide the integration domain to remove such discontinuities or to ensure any singularities are located at a vertex or along the boundary of the integration domain. *r2d2lri* is particularly robust in that its rate of failure to return the requested accuracy or to reliably report that the requested accuracy cannot be achieved is extremely low.

elrint3d has a similar structure to *r2d2lri* and is also written in C++. It is based on an 823-point seed lattice that was determined following an exhaustive search using goodness criteria (both Index of Merit and Trigonometric Degree of Precision) for the resulting augmentation sequence (see [5]). It also incorporates the sixth-order Sidi transformation and makes use of extrapolation in both the cubature approximation and the error estimate. It is effective for the same diverse range of integrand behaviours in three dimensions as is *r2d2lri* in two dimensions. Like *r2d2lri*, its non-adaptive nature makes it possible to pre-compute and store all of its abscissae and weights, the resulting low overhead rendering it extremely fast.

3 *plrint5d*

Essentially, a call to *plrint5d* can be seen as a call to *elrint3d* in which each three-dimensional function evaluation is performed via a call to *r2d2lri*. Specifically, we write the required five-dimensional integral as

$$If = \int_{D_3} f(x, y, z)\, dzdydx,$$

where

$$f(x, y, z) = \int_{D_2} g(x, y, z, u, v)\, dvdu \tag{1}$$

and $D_3 \times D_2$ is the overall integration domain. Then *elrint3d* is used to evaluate If, with *r2d2lri* being used to compute each of the values of f within *elrint3d*.

This model provides two interesting challenges:

1. With the maximum number of function evaluations for *elrint3d* and *r2d2lri* set at 421,376 and 22,784, respectively, the amount of computation required to evaluate a difficult five-dimensional integral is potentially extremely high. Despite the low overheads involved and the high speed of the component routines, a single integral evaluation could be very time-consuming.
2. A software interface problem arises in the setting of the tolerance parameters in each of *elrint3d* and *r2d2lri* in a way that most reliably achieves the required overall tolerance.

We deal with the first of these challenges using multi-threading. For the second, we introduce a simple but effective dynamic error control procedure.

4 Multi-threading in *plrint5d*

Netbook computers and PDAs aside, it is rare these days for new desktop or laptop computers to have a single core. Multi-core, multi-processor machines have rapidly become the norm and this means that algorithm designs must make optimal use of the parallel processing opportunities provided by these environments. *elrint3dmt* is an automatic cubature routine for three-dimensional integrals designed to do just that.

elrint3dmt is an improved version of *elrint3d* that uses a Shared-Memory Parallelism (SMP) design. It is the same as *elrint3d* in terms of its features and functionality, but its function evaluations are distributed synchronously to different threads, leaving the remaining computations such as error estimation, extrapolation and termination checks on the main thread. Users are provided with the option of specifying the number of threads.

Good scalability in such a simple strategy will result when three criteria are satisfied:

1. the number of threads matches the number of cores available;
2. a large number of function evaluations is required, and
3. each function evaluation is time-consuming.

Assuming the number of threads is chosen appropriately, the last two criteria are normally satisfied by *plrint5d*, especially when high accuracies are requested. The minimum number of function evaluations possible in *plrint5d* is 6,584; considerably more than this number will be needed for difficult integrals when medium-to-high accuracies are required. Furthermore, as each function call requires the evaluation of a two-dimensional integral, computation of a set of such function evaluations on a single thread is normally much more time-consuming than are the computations on the main thread required to implement the algorithm strategy. Accordingly, we use *elrint3dmt* in place of *elrint3d* in *plrint5d*.

The non-adaptive nature of the underlying algorithm in *elrint3d* makes implementation of the multi-threading strategy in *elrint3dmt* quite straightforward. Essentially, for a given integral, the algorithm proceeds one step at a time, computing the next lattice rule in the sequence, each time using pre-determined (and stored) weights and abscissae, with the resulting cubature approximation computed using that rule being compared with approximations computed from previous rules in the sequence. So, if we assume that *plrint5d* is run using n threads and there are N function evaluations in the next lattice rule to be evaluated, then for that evaluation N/n computations will be carried out in each thread (with appropriate adjustment if N is not divisible by n). In this way, all threads will return their computed weighted sum of function values at virtually the same time, the main thread will update the overall cubature approximation and error estimate, perform termination checks and the process will then be repeated if computation of another lattice rule in the sequence is required.

The time required for updating information on the main thread is normally minuscule in relation to the time taken for the function evaluations. Notwithstanding, multi-threading is not employed for the computation of the initial 823-point rule as the cost of implementing it will generally outweigh the time-saving to be gained otherwise.

Standard C++ does not provide a uniform standard thread library. Windows multi-threading applications are often implemented using WIN32 API (see [6] for details), while PThread [7] is used for multi-threading developments on a Linux platform. For *elrint3dmt*, multi-threading is realized using the Boost C++ libraries [8], a collection of peer-reviewed, open source libraries that extend the functionality of C++.

Boost.Thread enables the use of multiple threads of execution with shared data in portable C++ code. It provides classes and functions for managing the threads themselves, along with others for synchronizing data between the threads or providing separate copies of data specific to individual threads. In *elrint3dmt*, Class `boost::thread` is used for creating the threads for computing sets of function evaluations; class `boost::mutex::scoped_lock` is used for synchronizing the code block which adds the numbers of function evaluations returned by *r2d2lri*, and class `boost::thread_specific_ptr` is used for storing the dynamic error tolerance used in each thread (as described in the next section).

5 Error Control Strategy

Recalling that a call to *plrint5d* becomes a call to *elrint3dmt*, it would seem appropriate that the requested error in the call to *plrint5d* become the requested error in the call to *elrint3dmt* . However, the normal assumption when using *elrint3dmt* directly is that each function value used in the computation of the integral is calculated to close to machine precision. Satisfaction of this assumption in *plrint5d* would therefore dictate that the requested error in each call to *r2d2lri* be close to machine precision. But in some difficult cases, such a small requested error may not be achievable, causing *plrint5d* to fail even when only moderate accuracy is required in the overall cubature. Further, even if *r2d2lri* can achieve close to machine precision in each cubature, in cases where only low to moderate accuracy is required overall, the routine may be unnecessarily inefficient. So, we are faced with the question of how best to set the error tolerance parameters in the call to the outer routine (*elrint3dmt*) and in each call to the inner routine (*r2d2lri*) to most reliably satisfy a specified overall tolerance.

In [9], Fritsch et al consider a similar software interface problem for two-dimensional integration, but their approach is specific to the particular one-dimensional routines used. The problem is also considered in a more general context in [10].

Notwithstanding that numerical integration is an inherently unreliable process (see, for example, [11]), in what follows we assume that, unless it flags otherwise, *r2d2lri* is successful in achieving the requested accuracy in its integrations and that the error estimate it returns is also reliable. This is a reasonable assumption in practice; on the very rare occasions that *r2d2lri* fails to achieve the required accuracy without setting its error flag, it normally just fails to do so.

Let E be the required absolute error in the computation of If (that is, E is the requested error in the call to *plrint5d*). Also, let e_e be the requested error in the call to *elrint3dmt*. For the computation of a particular N-point cubature within *elrint3dmt*, let $e_{r_i},\ i = 1, \ldots, N$, be the requested error in the i^{th} call to *r2d2lri*. Further, let ε_e be the actual error in the N-point cubature within *elrint3dmt* and let ε_{r_i} be the actual error in the i^{th} cubature returned by *r2d2lri* for the calculation of that n-point rule.

We denote the N-point cubature computed by *elrint3dmt* by

$$Q\tilde{f} = \sum_{i=1}^{N} w_i \tilde{f}(\mathbf{x}_i),$$

where $\tilde{f}$ represents the approximation to the true function value f that is returned by *r2d2lri*. Then, noting that all the weights $w_i, i = 1, \ldots, N$ are positive, we have

$$
\begin{aligned}
|If - Q\tilde{f}| &= |If - \sum_{i=1}^{N} w_i(f(\mathbf{x}_i) + \varepsilon_{r_i})| \qquad (2)\\
&\le |If - Qf| + \sum_{i=1}^{N} w_i|\varepsilon_{r_i}|\\
&= |\varepsilon_e| + \sum_{i=1}^{N} w_i|\varepsilon_{r_i}|.
\end{aligned}
$$

Accordingly, if we denote the error estimate calculated by *elrint3dmt* for the current N-point cubature by $\tilde{e}_e$ and the error estimate returned by *r2d2lri* for the i^{th} function value by $\tilde{e}_{r_i}$, and assume that these error estimates are reliable indicators of the corresponding true errors, then an appropriate error estimate for the N-point cubature within *plrint5d* is

$$e = \tilde{e}_e + \sum_{i=1}^{N} w_i\tilde{e}_{r_i}. \qquad (3)$$

In practice, $\tilde{e}_e$ and $\tilde{e}_{r_i}$ are normally conservative upper bounds on the corresponding true errors, so (3) will normally also be a conservative error estimate, especially allowing for the possibility of cancellation of positive and negative values of the true errors in (2).

There remains the question of how best to set the tolerances e_e and e_{r_i} in a way to best achieve the overall tolerance E. In *plrint5d* , we set $e_e = (1-\alpha)E$ and $e_r = \alpha E$, with $\alpha = 0.1$, this value of α having been determined empirically. Note that e_r is a total error requirement across the function evaluations that are used by *elrint3dmt* in computing a single cubature. Provided $\tilde{e}_e \le e_e$ and $\tilde{e}_r = \sum_{i=1}^{N} w_i\tilde{e}_{r_i} \le e_r$, we will have $e \le E$, as required, where e is defined in (3).

Now suppose we wish to ensure (or, more precisely, aim to ensure) that each function value f is evaluated to a guaranteed minimum accuracy, η, i.e., each $\tilde{e}_{r_i} \le \eta, i = 1, \ldots, N$. Then $\tilde{e}_r \le \eta\sum_{i=1}^{N} w_i$ and therefore, in order that $\tilde{e}_r \le e_r$, we need to choose $\eta \le e_r/\sum_{i=1}^{N} w_i$. Since $\sum_{i=1}^{N} w_i = 1$ for each N in *elrint3dmt*, we thus require $\eta \le e_r$.

We take into account the conservative nature of the error estimates returned by *r2d2lri* by not automatically setting all e_{r_i} to e_r. Rather, we use the following adaptive error control strategy.

Set $e_{r_1} = e_r$. Then for $i = 2, \ldots, N$, set

$$
e_{r_i} = \begin{cases} e_{r_{i-1}}/10, & \tilde{e}_{r_{i-1}} \ge e_r/10\\ e_{r_{i-1}}, & e_r/100 \le \tilde{e}_{r_{i-1}} < e_r/10\\ 10 * e_{r_{i-1}}, & \tilde{e}_{r_{i-1}} < e_r/100. \end{cases}
$$

In *plrint5d*, this strategy is implemented independently in each thread. Since successive points at which the three-dimensional function f is calculated within each thread are contiguous in one dimension, it is assumed that the performance of *r2d2lri* varies little from integration $i-1$ to integration i. Thus, if the error estimate for integration $i-1$ is very close to the requested error, we tighten the requested error for the next integration. On the other hand, if the error estimate for the current integration indicates that the required accuracy has been well exceeded, then we can afford to ease the tolerance for the following integration. If neither of these conditions is satisfied, we leave the tolerance unchanged. In this way, it is hoped that the required accuracies can be achieved reliably while minimizing the number of evaluations of the function g in (1) during the calls to *r2d2lri*.

Provided the error estimates can be trusted, this strategy ensures that the error in every function evaluation is less than η and therefore that $\tilde{e}_r \le e_r$, as required.

Normal termination of *plrint5d* occurs when $e \le E$. Abnormal termination occurs if any of the usual abnormal termination conditions in *elrint3dmt* occur (such as the maximum number of function evaluations being reached or rounding error preventing the requested accuracy being achieved), but it is also necessary to take into account the performance of *r2d2lri*. If, for a particular N, it is found that $\sum_{i=1}^{N} w_i \tilde{e}_{r_i} > E$, then it is assumed that no further improvement can be gained and *plrint5d* is terminated. It is also possible to set a maximum number of function evaluations for *plrint5d* itself.

6 Use of *plrint5d*

To use *plrint5d*, the integrand must be provided either as a function pointer or as a class that extends the base class `Integrand<double>`. The integration domain can be defined using constants or function pointers. When the integration domain is a five-dimensional hyper-rectangle, although function pointers can be used to describe the domain, users are strongly recommended to use constants since the routine has in-built efficiencies in such a case. Infinite regions are specified using the supplied constant INFINITY. The user may supply an absolute and/or relative tolerance; if both are specified, the routine will attempt to satisfy the easier of the two.

The following example demonstrates how *plrint5d* could be used to evaluate the integral

$$I_s = \int_{[0,1]^5} \sqrt{x_4 x_5 (1 - x_1 x_2 x_3)}\, d\mathbf{x}. \tag{4}$$

In this example, the relative tolerance is set to 10^{-8} and the number of threads is set to 4.

```
#include   <iostream>
#include   <plrint5d.h>

using namespace std;
```

```
double sample_f(double x1, double x2, double x3, double x4, double x5)
{
   return sqrt(x4*x5*(1 - x3*x4*x5));
}

int main(int argc, char** argv)
{
   Plrint5d integ(sample_f, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1e-8);

   integ.setThreads(4);

   cout << "Cubature = " << integ.evaluate() << endl;
   cout << "Estimated Error = "<< integ.estErr() << endl;
   cout << "Error Flag = "<< integ.errFlag() << endl;
   cout << "No of Function Values = " << integ.productEvals()
        << endl;

   return 0;
}
```

For the integral

$$\int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^{1-x_1} \sqrt{x_4 x_5 (1 - x_1 x_2 x_3)}\, dx_5 dx_4 dx_3 dx_2 dx_1,$$

we use functions to define the integration domain, as follows.

```
double a2 (double x1)
{
   return 0.0;
}

double b2 (double x1)
{
   return 1.0;
}

double a3 (double x1, double x2)
{
   return 0.0;
}

double b3 (double x1, double x2)
{
   return 1.0;
}

double a4 (double x1, double x2, double x3)
{
   return 0.0;
}

double b4 (double x1, double x2, double x3)
{
   return 1.0;
}

double a5 (double x1, double x2, double x3, double x4)
{
   return 0.0;
}

double b5 (double x1, double x2,  double x3, double x4)
{
   return 1.0 - x1;

}
```

Then the call to *plrint5d* takes the form

```
Plrint5d integ(sample_f, 0, 1, a2, b2,a3, b3, a4, b4, a5, b5, 1e-8);
```

A class is used in the following example from the Gaussian family (see (8) in Appendix A), first proposed by Genz [12].

```
class GaussInt : public Integrand<double>
{
private:
   double c[5];
   double w[5];

public:
   GaussInt(double c1, double c2, double c3, double c4, double c5,
            double w1, double w2, double w3, double w4, double w5)
   {
      c[0] = c1;  c[1] = c2; c[2] = c3;  c[3] = c4; c[4] = c5;
      w[0] = w1;  w[1] = w2; w[2] = w3;  w[3] = w4; w[4] = w5;
   }

   double fun(const double x[])
   {
      return
      exp(- pow(c[0]*(x[0] - w[0]), 2) - pow(c[1]*(x[1] - w[1]), 2)
          - pow(c[2]*(x[2] - w[2]), 2) - pow(c[3]*(x[3] - w[3]), 2)
          - pow(c[4]*(x[4] - w[4]), 2));
   }
};
```

```
int main(int argc, char** argv)
{
   GaussInt integ(0.1, 0.2, 0.6, 0.8, 0.9, 1.0, 1.0, 1.0, 1.0, 1.0);
   Plrint5d integ(&integ, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1e-8);
   ...
}
```

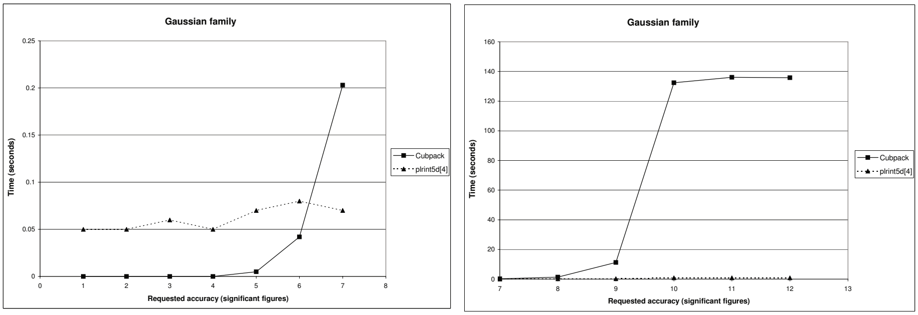
7 Numerical Examples

To demonstrate the performance of *plrint5d*, we have tested it on the six Genz families [12] (see Appendix A), an integral over an infinite region and a number of integrals involving algebraic singularities on one or more boundaries of the integration region. With the exception of the infinite range integral, we have also computed each of these integrals using the highly-respected Cubpack routine [13], an adaptive routine for general-purpose multi-dimensional integration. Notwithstanding that the FORTRAN compiler gFORTRAN has been used to compile Cubpack and Visual Studio C++ has been used for compiling *plrint5d*, we present some data that exemplifies the excellent performance of *plrint5d* when medium to high accuracies are required and particularly when boundary singularities are involved. All tests were carried out on a 3Ghz quad-core machine with 3GB of RAM.

The relative performances of *plrint5d* and Cubpack follow a similar pattern for each of the Genz Families 1–4. In Figures 1a and 1b, we use Family 4 (Gaussian) as representative of these results and show the time taken to achieve each of the requested accuracies 10^{-n}, $n = 1, 2, \ldots, 12$ using Cubpack and *plrint5d* with four threads. As can be seen in Figure 1a, Cubpack is clearly faster and much preferred for accuracies up to 10^{-6}. However, the picture changes dramatically as higher accuracies are required. For accuracies beyond 10^{-9}, Cubpack reaches the preset maximum number of function values (100,000,000) without achieving the desired result, the time for the maximum number of evaluations therefore remaining level across the range of high requested accuracies in Figure 1b. By contrast, *plrint5d* achieves all accuracies up to 10^{-12} several orders of magnitude faster than Cubpack. (See also the Appendix for more detailed results for this family.)

plrint5d is not at all suited to Families 5–6, which involve discontinuities in the integrand or its derivative, but it is nevertheless robust in efficiently returning a failure flag for these integrals when the requested accuracy cannot be achieved. In practice, one would normally subdivide the integration region to eliminate the discontinuities and render the integrations straightforward.

The performance of *plrint5d* for an integral with algebraic boundary singularities is depicted in Figure 2 for the integral I_s defined in (4). In this case, Cubpack is to be preferred only for very low accuracies (10^{-1} and 10^{-2}). *plrint5d* achieves all accuracies very efficiently. Cubpack reaches the preset maximum number of function evaluations for tolerances beyond 10^{-6}.



(a) Low accuracy (b) High accuracy

Fig. 1. Time comparison for Family 4

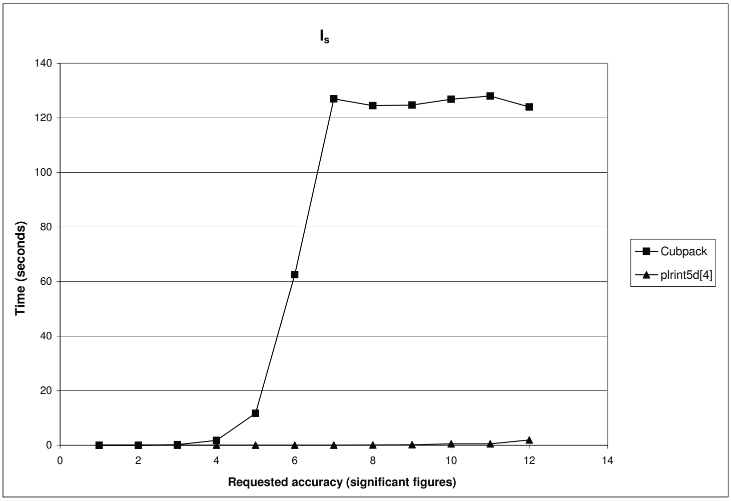


Fig. 2. Time comparison for I_s

The scalability of the multi-threading strategy in *plrint5d* is exemplified in Figures 3 and 4. The points on the graphs represent the ratio of times taken using two and four threads compared to the time taken using one thread. Shown on each graph are the speed-ups achieved when 4-digit accuracy is requested and when 10-digit accuracy is requested. The “perfect speed-up” line is included for reference. As is supported by these graphs, for high accuracies, and particularly for difficult integrals, the improvement gained in using more threads is not too far from optimal (provided a sufficient number of cores is available).

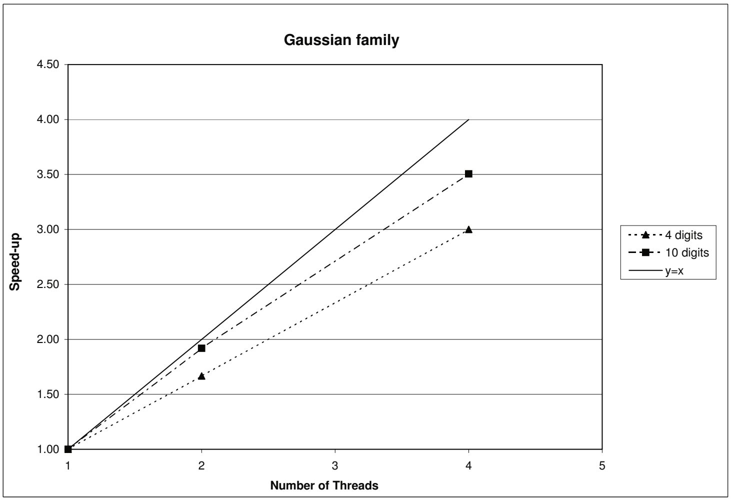


Fig. 3. Speed scaling for Family 4

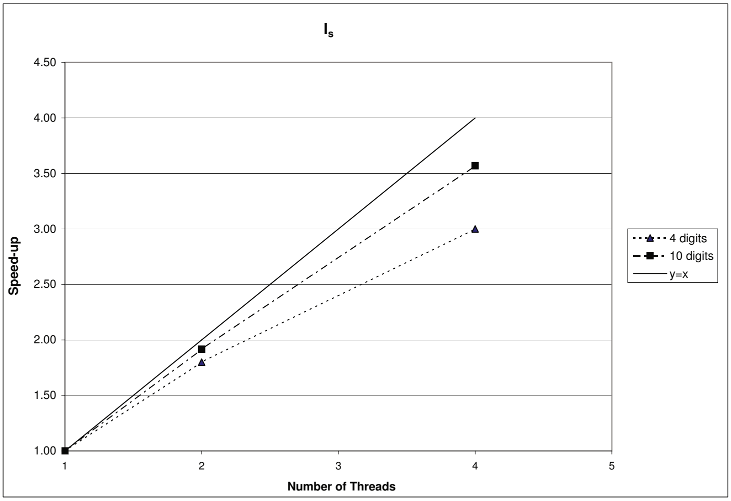


Fig. 4. Speed scaling for Family I_s

8 Conclusion

plrint5d proves to be an effective and very fast routine for computing five-dimensional integrals, especially when medium to high accuracies are required and/or when the integrand has certain forms of singularity on the boundary of the integration domain. Its design allows integrals with variable boundaries and integrals over infinite regions to be evaluated.

The multi-threading strategy used in *plrint5d* is particularly effective in offsetting the expense of computing a large number of function values. Tests to date have been limited to a quad-core machine, but the strategy employed ensures that the excellent speed-up demonstrated here is likely to be maintained for a larger number of cores.

The approach employed in *plrint5d* could easily be applied in the evaluation of four-dimensional and six-dimensional integrals using *r2d2lri* as both the inner and outer routine in the four-dimensional case and *elrint3d* as the inner and outer routine in the six-dimensional case. In this way, *r2d2lri* and *elrint3d* together provide the basis of an excellent suite of routines for computing two-dimensional to six-dimensional integrals.

A Genz Test Families

Listed below are the Genz test families. $\mathbf{w} = (w_1, w_2, w_3, w_4, w_5)$ and $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5)$ are randomly generated vectors in $[0,1]^5$. (See [12] for additional details.)

1. Oscillatory

$$\int_0^1\int_0^1\int_0^1\int_0^1\int_0^1 \cos(2\pi w_1 + \sum_{i=1}^{5} c_i x_i) dx_5 dx_4 dx_3 dx_2 dx_1 \tag{5}$$

2. Product Peak

$$\int_0^1\int_0^1\int_0^1\int_0^1\int_0^1 \frac{1}{\prod_{i=1}^{5}(c_i^2 + (x_i - w_i)^2)} dx_5 dx_4 dx_3 dx_2 dx_1 \tag{6}$$

3. Corner Peak

$$\int_0^1\int_0^1\int_0^1\int_0^1\int_0^1 \frac{1}{(1 + \sum_{i=1}^{5} c_i x_i)^6} dx_5 dx_4 dx_3 dx_2 dx_1 \tag{7}$$

4. Gaussian

$$\int_0^1\int_0^1\int_0^1\int_0^1\int_0^1 e^{-\sum_{i=1}^{5} c_i^2 (x_i - w_i)^2} dx_5 dx_4 dx_3 dx_2 dx_1 \tag{8}$$

5. Discontinuous first derivative

$$\int_0^1\int_0^1\int_0^1\int_0^1\int_0^1 e^{-\sum_{i=1}^{5} c_i |x_i - w_i|} dx_5 dx_4 dx_3 dx_2 dx_1 \tag{9}$$

6. Discontinuous

$$\int_0^1\int_0^1\int_0^1\int_0^1\int_0^1 f_6(x_1, x_2, x_3, x_4, x_5) dx_5 dx_4 dx_3 dx_2 dx_1, \tag{10}$$

where

$$f_6(x_1, x_2, x_3, x_4, x_5) = \begin{cases} e^{\sum_{i=1}^{5} c_i x_i}, & \text{if } x_i < w_i, \\ 0, & \text{otherwise.} \end{cases}$$

B Sample Results

In the following tables,

ReqRelErr = requested relative error,
ActRelErr = actual relative error,
EstRelErr = estimated relative error,
FunEvals = number of function evaluations,
ErrFlag = error indication flag, and
Time = real time in seconds.

Table 1. Results for Family 4 using *plrint5d* with 4 threads

ReqRelErr	ActRelErr	EstRelErr	FunEvals	ErrFlg	Time
1.00E-01	2.50E-07	8.50E-06	1744760	0	0.05
1.00E-02	2.50E-07	8.50E-06	1744760	0	0.05
1.00E-03	2.50E-07	8.50E-06	1744760	0	0.06
1.00E-04	2.50E-07	8.50E-06	1744760	0	0.05
1.00E-05	2.80E-15	5.60E-09	2324152	0	0.07
1.00E-06	2.80E-15	5.60E-09	2324152	0	0.08
1.00E-07	2.80E-15	5.60E-09	2324152	0	0.07
1.00E-08	4.70E-13	2.20E-09	6998792	0	0.20
1.00E-09	3.20E-14	8.10E-10	9329528	0	0.27
1.00E-10	1.30E-16	6.60E-14	37318112	0	0.86
1.00E-11	1.30E-16	6.60E-14	37318112	0	0.86
1.00E-12	1.30E-16	6.60E-14	37318112	0	0.87

Table 2. Results for Family 4 using *Cubpack*

ReqRelErr	ActRelErr	EstRelErr	FunEvals	ErrFlg	Time
1.00E-01	6.94E-05	1.43E-04	103	0	0.00
1.00E-02	6.94E-05	1.43E-04	103	0	0.00
1.00E-03	6.94E-05	1.43E-04	103	0	0.00
1.00E-04	3.05E-05	3.70E-05	309	0	0.00
1.00E-05	9.29E-07	9.55E-06	3605	0	0.01
1.00E-06	8.23E-09	9.97E-07	30591	0	0.04
1.00E-07	3.42E-11	9.99E-08	150483	0	0.20
1.00E-08	5.49E-11	1.00E-08	1012181	0	1.35
1.00E-09	9.03E-13	1.00E-09	8375239	0	11.22
1.00E-10	9.93E-13	6.56E-10	99999919	1	132.41
1.00E-11	9.93E-13	6.56E-10	99999919	1	136.08
1.00E-12	9.93E-13	6.56E-10	99999919	1	135.82

Table 3. Results for $I_s = \int_{[0,1]^5} \sqrt{x_4 x_5 (1 - x_1 x_2 x_3)}\, d\mathbf{x}$ using *plrint5d* with 4 threads

ReqRelErr	ActRelErr	EstRelErr	FunEvals	ErrFlg	Time
1.00E-01	3.50E-09	3.20E-08	1744760	0	0.03
1.00E-02	3.50E-09	3.20E-08	1744760	0	0.03
1.00E-03	3.50E-09	3.20E-08	1744760	0	0.04
1.00E-04	3.50E-09	3.20E-08	1744760	0	0.03
1.00E-05	3.50E-09	3.20E-08	1744760	0	0.04
1.00E-06	3.50E-09	3.20E-08	1744760	0	0.03
1.00E-07	5.80E-12	9.40E-09	2324152	0	0.05
1.00E-08	5.80E-12	1.30E-09	4668056	0	0.08
1.00E-09	5.80E-12	6.20E-10	9329528	0	0.17
1.00E-10	8.00E-16	8.80E-12	37318112	0	0.51
1.00E-11	8.00E-16	8.80E-12	37318112	0	0.52
1.00E-12	2.70E-16	2.20E-14	49272448	0	1.93

Table 4. Results for $I_s = \int_{[0,1]^5} \sqrt{x_4 x_5 (1 - x_1 x_2 x_3)}\, d\mathbf{x}$ using using *Cubpack*

ReqRelErr	ActRelErr	EstRelErr	FunEvals	ErrFlg	Time
1.00E-01	1.04E-04	7.55E-02	515	0	0.00
1.00E-02	8.33E-05	4.54E-03	1133	0	0.01
1.00E-03	2.84E-05	9.99E-04	187151	0	0.23
1.00E-04	7.56E-06	1.00E-04	1463733	0	1.80
1.00E-05	4.26E-07	1.00E-05	9378665	0	11.71
1.00E-06	2.15E-08	1.00E-06	49864463	0	62.51
1.00E-07	6.59E-09	3.86E-07	99999919	1	126.98
1.00E-08	6.59E-09	3.86E-07	99999919	1	124.50
1.00E-09	6.59E-09	3.86E-07	99999919	1	124.72
1.00E-10	6.59E-09	3.86E-07	99999919	1	126.85
1.00E-11	6.59E-09	3.86E-07	99999919	1	128.02
1.00E-12	6.59E-09	3.86E-07	99999919	1	124.01

References

1. Robinson, I., Hill, M.: An algorithm for automatic two-dimensional cubature. ACM Transactions on Mathematical Software 28(1), 73–89 (2002)
2. Li, T., Robinson, I.: elrint3d: A three-dimensional non-adaptive automatic cubature routine using a sequence of embedded lattice rules. Submitted for publication
3. Zaremba, S.K.: Good lattice points, discrepancy and numerical integration. Annali di Matematica Pura ed Applicata 73, 293–317 (1966) (In Italian)
4. Sidi, A.: A new variable transformation for numerical integration. International Series of Numerical Mathematics 112, 359–373 (1993)
5. Li, T., Robinson, I.: The search for a good lattice augmentation sequence in three dimensions. In: Gervasi, O., Gavrilova, M.L. (eds.) ICCSA 2007, Part III. LNCS, vol. 4707, pp. 1037–1045. Springer, Heidelberg (2007)

6. Beveridge, J., Wiener, R.: Multithreading Applications in Win32: The Complete Guide to Threads. Addison-Wesley Professional, Reading (1996)
7. Lewis, B., Berg, D.J.: Multithreaded programming with Pthreads. Prentice-Hall, Inc, Englewood Cliffs (1998)
8. Abrahams, D., Gurtovoy, A.: C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond. Addison-Wesley Professional, Reading (2004)
9. Fritsch, F.N., Kahaner, D.K., Lyness, J.N.: Double integration using one-dimensional adaptive quadrature routines: A software interface problem. ACM Transactions on Mathematical Software 7(1), 46–75 (1981)
10. Kahaner, D., Moler, C., Nash, S.: Numerical Methods and Software. Prentice-Hall, Englewood Cliffs (1988)
11. Lyness, J.N., Kaganove, J.: Comments on the nature of automatic quadrature routines. ACM Transactions on Mathematical Software 2(1), 65–81 (1976)
12. Genz, A.C.: Testing multidimensional integration routines. In: Ford, B., Rault, J.C., Thomasset, F. (eds.) Tools, Methods and Languages for Scientific and Engineering Computation, pp. 81–94. North-Holland, Amsterdam (1984)
13. Cools, R., Haegemans, A.: Algorithm 824: Cubpack: a package for automatic cubature; framework description. ACM Transactions on Mathematical Software 29(3), 287–296 (2003)

Development of Quadruple Precision Arithmetic Toolbox QuPAT on Scilab

Tsubasa Saito[1], Emiko Ishiwata[2], and Hidehiko Hasegawa[3]

[1] Graduate School of Science, Tokyo University of Science, Japan
[2] Tokyo University of Science, Japan
[3] University of Tsukuba, Japan

Abstract. When floating point arithmetic is used in numerical computation, cancellation of significant digits, round-off errors and information loss cannot be avoided. In some cases it becomes necessary to use multiple precision arithmetic; however some operations of this arithmetic are difficult to implement within conventional computing environments. In this paper we consider implementation of a quadruple precision arithmetic environment QuPAT(Quadruple Precision Arithmetic Toolbox) using the interactive numerical software package Scilab as a toolbox. Based on Double-Double(DD) arithmetic, QuPAT uses only a combination of double precision arithmetic operations. QuPAT has three main characteristics: (1) the same operator is used for both double and quadruple precision arithmetic; (2) both double and quadruple precision arithmetic can be used at the same time, and also mixed precision arithmetic is available; (3) QuPAT is independent of which hardware and operating systems are used. Finally we show the effectiveness of QuPAT in the case of analyzing a convergence property of the GCR(m) method for a system of linear equations.

Keywords: quadruple precision arithmetic, mixed precision, Scilab.

1 Introduction

Floating point arithmetic operations governed by IEEE754 are mainstream on conventional computers. But in floating point arithmetic, we cannot avoid cancellation of significant digits, round-off errors and information loss. In the case of double precision floating point numbers, there are approximately 16 (decimal) significant digits. Therefore, when a system of linear equations is solved by some iterative method, it is known that stagnation of the residual norm may occur or that the numerical solution may not converge. To reduce these errors, we need to use multiple precision arithmetic. However, it is difficult to implement multiple precision arithmetic in ordinary computing environments without any special hardware.

We consider quadruple precision arithmetic as multiple precision arithmetic. There is a proposal called Double-Double(DD) for quadruple precision arithmetic. DD is based on the algorithm for error-free floating point arithmetic by Dekker[2] and Knuth[5]. A DD number is represented by two double precision floating point numbers. QD[1] and Lis[6] are implementations of DD arithmetic in C and C++. In addition, although it is not standard, many Fortran compilers have a quadruple precision real number type 'REAL(KIND=16)' together with operations on such. The computational cost is high,

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 60–70, 2010.

however, if we cannot use special hardware. On the other hand, if we do use it, then code should be rewritten to use quadruple precision arithmetic hardware or library (except for Fortran). The rewritten code does not execute in the previous environment (i.e. without any special hardware or library), and rewritten code is difficult to debug.

However, there does exist an interactive numerical software package Scilab[10]. Scilab is similar to MATLAB, and, moreover, this is free and open source software. In this paper we implement a quadruple precision arithmetic environment QuPAT (Quadruple Precision Arithmetic Toolbox) using the interactive numerical software package Scilab as a toolbox.

In Scilab, we can define a new data type and apply operator overloading. Thus, we define a new data type representing a DD number to expand the double precision arithmetic environment. Using operator overloading, we can use the same operator for both double precision arithmetic and quadruple precision arithmetic. We can also use both double precision arithmetic and quadruple precision arithmetic at the same time, and thus we can make use of mixed precision arithmetic. QuPAT is implemented only using Scilab functions, so that QuPAT is independent of hardware and operating systems. Therefore, Scilab users can easily make use of QuPAT anywhere that it is required.

This paper is organized as follows. Section 2 presents some algorithms for error-free double precision floating point arithmetic and DD arithmetic. In Section 3, we describe the way to construct a DD environment on Scilab, the characteristics of QuPAT, and the computational time for DD arithmetic. In Section 4 we show the effectiveness of QuPAT for analyzing a convergence property of the GCR(m) method for a system of linear equations. Section 5 presents a summary and discusses future work.

2 DD Arithmetic

We first explain DD arithmetic, which is based on representation using two double precision floating point numbers and defining four error-free floating point arithmetic algorithms [2], [5]. Specifically we explain the characteristics and four arithmetic operations of DD.

2.1 Characteristics of DD Arithmetic

A DD number is represented using two double precision floating point numbers. A real number α is represented as the DD number $A = (Ahi, Alo)$, which is defined below:

$$Ahi = (\ \alpha \text{ rounded to a double precision number})$$
$$Alo = (\ (\alpha - Ahi) \text{ rounded to a double precision number})$$

Figure 1 shows the constitution of each DD number in bits and an IEEE754 quadruple precision floating point number. A DD thus contains two sign bits and a pair of 11 bit sequences for the exponent parts. But the sign and the exponent part of a DD number depend only on Ahi. The mantissa of a DD number contains in total 104 bits. This is 8 bits less than is required for an IEEE quadruple precision number. A double precision number d can be transformed to a DD number by setting $Ahi = d$, $Alo = 0$. For arithmetical operations on DD numbers, if the computational result doesn't fit into a DD number, then it is rounded to a DD number.

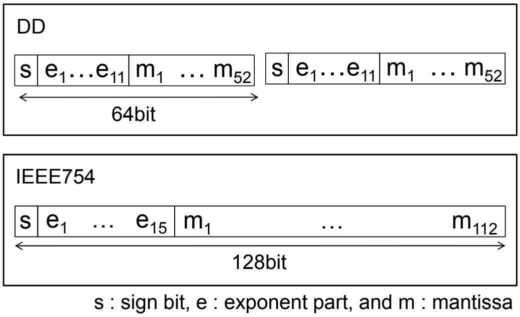


Fig. 1. Constitution of bits (DD number and IEEE754 quadruple precision number)

2.2 Algorithm for DD Arithmetic

For implementation of DD arithmetic, double precision arithmetic should be computed exactly. However, a result of double precision arithmetic does not always fit into a double precision number. Therefore, we need to express such a number correctly using two double precision numbers. $f(\quad\cdot\quad)$ denotes a computation of double precision arithmetic, variables of small letter are double precision numbers, variables of capital letter are DD numbers. We present four algorithms (D1)‘(D4) for error-free double precision arithmetic below (see [3], [4] for details).

Algorithm (D1) - two_sum
The following algorithm computes double precision addition $s = f(a + b)$ and error $e = (a + b) - s$. In this way, the sum of two double precision floating point numbers is represented strictly: $a + b = s + e$.

two_ sum

```
$[s, e] = two\_sum\ (a, b)$
    $s = a + b;$
    $v = s - a;$
    $e = (a - (s - v)) + (b - v);$
$end$
```

Algorithm (D2) - fast_two_sum (for $|a| \geq |b|$)
The following algorithm is almost the same as (D1) except for the assumption that $|a| \geq |b|$.

fast_two_sum

```
$[s, e] = fast\_two\_sum\ (a, b)$
    $s = a + b;$
    $e = b - (s - a);$
$end$
```

Algorithm (D3) - split

The following algorithm splits a double precision floating point number a into h and l where h contains the higher 26 bits of the mantissa of a, and l contains the lower 26 bitsD

split

```
$[h, l] = split\ (a)$
    $t = 134217729 * a;$
    $h = t - (t - a);$
    $l = a - h;$
$end$
```

Algorithm (D4) - two_prod

The following algorithm computes double precision multiplication $p = f(a \times b)$ and error $e = (a \times b) - p$. Using (D3), we get the following equality and algorithm :

$$a \times b = f(ah \times bh) + f(ah \times bl) + f(al \times bh) + f(al \times bl)$$

two_ prod

```
$[p, e] = two\_prod\ (a, b)$
    $p = a * b;$
    $(ah, al) = split\ (a);$
    $(bh, bl) = split\ (b);$
    $e = ((ah * bh - p) + ah * bl + al * bh) + al * bl;$
$end$
```

The four arithmetic operations of DD are defined only using double precision arithmetic numbers and their computation algorithms (D1)‘(D4). Let DD numbers A, B and C be (Ahi,Alo), (Bhi,Blo) and (Chi,Clo) , respectively.

Algorithm (DD1) - addition

The following algorithm computes DD addition $A + B$.

addition

```
$C = add\ (A, B)$
    $[sh, eh] = two\_sum\ (Ahi, Bhi);$
    $[sl, el] = two\_sum\ (Alo, Blo);$
    $se = eh + sl;$
    $[sh', se'] = fast\_two\_sum\ (sh, se);$
    $see = se' + el;$
    $[Chi, Clo] = fast\_two\_sum\ (sh', see);$
$end$
```

Algorithm (DD2) - subtraction
The following algorithm computes DD subtraction $A - B$ using DD addition.

subtraction

```
$C = sub\,(A, B)$
    $B^{*}hi = -Bhi$;
    $B^{*}lo = -Blo$;
    $C = add\,(A, B^{*})$;
$end$
```

Algorithm (DD3) - multiplication
The following algorithm computes DD multiplication $A \times B$ using the equality:

$$A \times B = Ahi \times Bhi + Ahi \times Blo + Alo \times Bhi + Alo \times Blo$$

However, we omit $Alo \times Blo$ from the computation to reduce the computation cost.

multiplication

```
$C = mul\,(A, B)$
    $[p1, p2] = two\_prod\,(Ahi, Ahi)$;
    $p2 = p2 + Ahi * Blo$;
    $p2 = p2 + Alo * Bhi$;
    $[Chi, Clo] = fast\_two\_sum\,(p1, p2)$;
$end$
```

Algorithm (DD4) - division
The following algorithm computes DD division ($A \div B$), assuming $B \neq 0$. DD division is based on Newton's method with an initial value $f(Ahi/Bhi)$.

division

```
$C = div\,(A, B)$
    $c = Ahi/Bhi$;
    $[p, e] = two\_prod\,(c, Bhi)$;
    $cc = (Ahi - p - e + Alo - c * Blo)/Bhi$;
    $[Chi, Clo] = fast\_two\_sum\,(c, cc)$;
$end$
```

Table 1 shows the number of double precision operations in the above algorithms, and Figure 2 shows the relationships among these algorithms.

Table 1. Number of double precision operations

	number of each operation			
	add & sub	mul	div	total
fast_two_sum	3	0	0	3
two_sum	6	0	0	6
split	3	1	0	4
two_prod	10	7	0	17
add	20	0	0	20
sub	20	0	0	20
mul	15	9	0	24
div	17	8	2	27

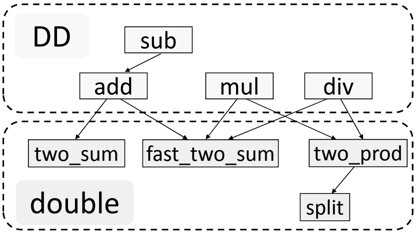


Fig. 2. Relationships among exact double precision arithmetic algorithms and DD algorithms

3 Construction of DD Environment on Scilab

Matlab and Scilab[10] are popular software packages for interactive numerical computation. They use double precision arithmetic, which can result in cancellation of significant digits, round-off errors and information loss. Scilab has almost the same capability as Matlab; however Scilab is open source and free. Scilab has been developed at Institut National de Recherche en Informatique et en Automatique (INRIA) in France.

In this paper, using DD, we construct a new environment for quadruple precision arithmetic QuPAT to reduce numerical errors, using Scilab as a toolbox.

3.1 Definition of Data Type for DD

Using Scilab, we can define a new data type using the Scilab function ‘tlist’. The function tlist creates a Scilab object and describes it as ‘tlist(typ, a1, ..., an)’, where ‘typ’ is our chosen name for the data type, and ‘a1, ..., an’ are elements. The values of a new data type are classified by its name. In this way we treat a data type using a combination of some data as a class in C++.

In Scilab, double precision numbers are defined by the data type named ‘constant’. Then, we define a new data type named ‘dd’ to contain DD numbers. To generate a value a as an element of the data type ‘dd’, we use the following code:

$$a = \text{tlist ([‘dd’,‘hi’,‘lo’], ahi, alo).}$$

Thus, ‘[‘dd’,‘hi’,‘lo’]’ represents the name of the data type and its elements, and ‘ahi’ and ‘alo’ are double precision values in the constant data type. The name of the new data type is ‘dd’. To refer to the value of the higher (resp. lower) part ahi (resp. alo), we simply type ‘a.hi’ (resp. ‘a.lo’).

In QuPAT, we define the following function to generate a DD number :

```
function a = dd(ahi,alo)
  a = tlist(['dd','hi','lo'],ahi,alo);
endfunction
```

For example, we define ‘a = dd(1,0)’, then the valuable of dd type a becomes 1.

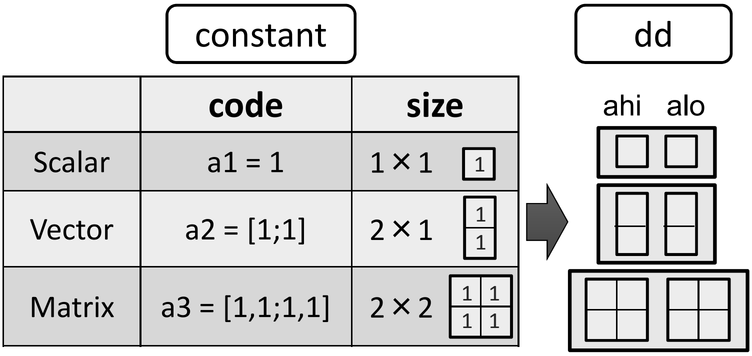


Fig. 3. Relationship between ‘constant’ and ‘dd’

If we transform a value *a* from ‘constant’ into ‘dd’, we may use the function ‘d2dd’. On the other hand, if we transform a value *a* from ‘dd’ into ‘constant’, we may refer the variable ‘a.hi’.

In Scilab, scalars, vectors and matrices are treated in the same way as the data type ‘constant’ (Figure 3). Then, defining only the data type ‘dd’ enables elements to be expanded into DD numbers naturally.

3.2 Definition of Operators for DD

On Scilab, we can make use of operator overloading for a new data type defined by tlist. For computing values of a new data type, Scilab calls the preliminarily defined function. Allowing operator overloading for computing a value of ‘dd’ using (DD1)‘(DD4), we can use the same operator for quadruple precision arithmetic as double precision arithmetic. In particular, we only redefine functions named ‘%<first_ operand_ type>_<op_code>_<second_operand_type>’ or ‘ %<operand_type>_<op_code>’. We need to write the sequence of characters associated with each data type to ‘<first_ operand_ type>’ and ‘<second_ operand_type>’. Similarly, we should write a single character associated with each operator to ‘<op_code>’.

For example, a single character ‘a’ can be assigned to the operator ‘+’. For computing the sum of DD numbers *A* and *B* using the operator ‘+’, we only define the function named %dd_a_dd. In the same way, computing the sum of a DD number *A* and a double precision number *b* using the operator ‘+’, we define the function named %dd_a_s, where ‘s’ is the character associated with the data type constant. For computation of mixed precision arithmetic with ‘dd’ and ‘constant’, the value of ‘constant’ is expanded to ‘dd’ in the computation.

3.3 Definition of a Function for DD

If a Scilab function is applied to an argument whose data type is ‘dd’ , then an error occurs. Hence, we define new functions named ‘dd<function_name>’ for calling with the type dd. For example, we create a new function ‘ddsqrt’ to be applied to a variable of type ‘dd’, corresponding to the Scilab function ‘sqrt’ to compute a square root.

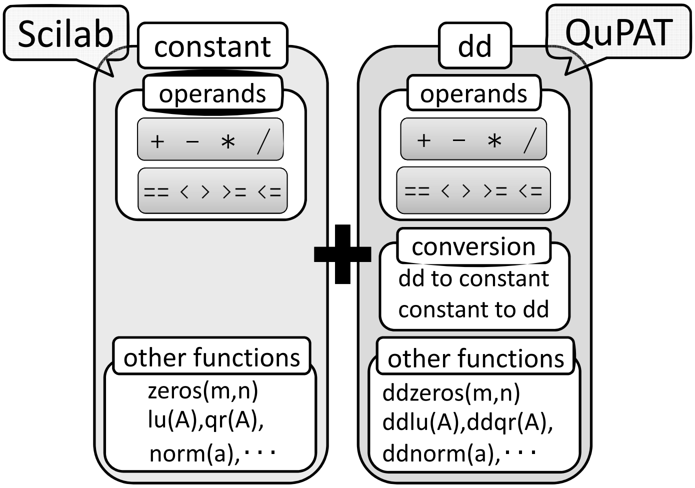


Fig. 4. Relationship between Scilab and QuPAT

3.4 Characteristics of QuPAT

In QuPAT (Quadruple Precision Arithmetic Toolbox), the data types ‘dd‘ and ‘constant’ are defined separately. Thus we can use both double precision arithmetic and quadruple precision arithmetic in the same code. In addition, we can apply mixed precision arithmetic using the same operator $(+, -, *, /)$ with QuPAT. To convert a value of type ‘constant’ into one of type ‘dd’, we assign 0 to the lower part of dd. In contrast, to convert a value of ‘dd’ into ‘constant’, we extract the higher part of dd. As a result, users of Scilab can use quadruple precision arithmetic with QuPAT in the same way as ordinary Scilab double precision arithmetic. Figure 4 shows the relationship between types ‘constant’ and ‘dd’ in Scilab. QuPAT is implemented using pure Scilab functions, hence QuPAT is independent of hardware and operating systems.

3.5 Computational Time for DD Arithmetic

Computation was carried out on a PC with an AMD Turion(tm) X2 Dual-Core 2.00GHz and Scilab version 5.1.1. Table 2 shows the computation time in seconds and the ratio of time required for DD arithmetic to time required for double precision arithmetic. Each result is the average over five trials, and N is the number of repetitions in the loop.

Computation time for DD arithmetic is about 9 to 15 times greater than that for double precision arithmetic.

4 Effectiveness of DD Arithmetic for Analyzing the GCR(m) Method

The GCR (Generalized Conjugate Residual) method is one of the the Krylov subspace methods to solve a nonsymmetric linear system $A\boldsymbol{x} = \boldsymbol{b}$. The GCR method is based on

Table 2. Computation time in seconds; the ratio is in parentheses

		computation time and ratio			
	N	addition	subtraction	multiplication	division
double	50,000	1.02	1.21	1.13	1.28
	100,000	2.08	1.92	1.98	1.95
	500,000	9.86	9.78	10.72	9.88
	1,000,000	19.24	20.11	19.69	19.25
DD	50,000	9.80 (9.62)	10.72 (8.87)	13.25 (11.74)	13.41 (10.49)
	100,000	19.91 (9.58)	21.12 (10.97)	28.00 (14.11)	29.00 (14.85)
	500,000	104.77 (10.63)	108.42 (11.08)	137.38 (12.82)	141.06 (14.28)
	1,000,000	207.74 (10.80)	218.23 (10.85)	278.83 (14.16)	280.11 (14.55)

Arnoldi process and the minimal residual approach. In addition, the GCR method has the theoretical property that the residual norm decreases at each iteration and converges after at most n iterations, where n is the dimension of the matrix A. But using floating point arithmetic, it is known that stagnation of the residual norm may occur and that the numerical solution may not converge.

In this section, we investigate numerical solution by the GCR(m) method where m is the restart cycle, comparing the results of using double precision arithmetic versus DD arithmetic. The difference in the code between double precision arithmetic and DD arithmetic is only in the definition of the variables and the name of the functions to compute a norm and an inner product. All experiments were carried out in the same computational environment as in Section 3.5.

We consider ‘arc30’ for the matrix A from the MatrixMarket [7]. The dimension of this matrix is $n = 130$, and its condition number is 6.05×10^{10}, obtained using the Scilab function ‘cond’. We set the restart cycle at $m = 50$, and GCR(m) in double and DD arithmetic was terminated at 1000 iterations if convergence did not occur. The iteration was started with $\boldsymbol{x}_0 = \boldsymbol{0}$ and the right-hand side vector $\boldsymbol{b}$ was given by substituting the solution vector $\boldsymbol{x}^* = (1, 1, ..., 1)^T$ into $\boldsymbol{b} = A\boldsymbol{x}^*$. The stopping criteria are given below:

$$||\boldsymbol{r}_k||_2 \leq 10^{-12}\,||\boldsymbol{r}_0||_2 \qquad \text{(for double)}$$

$$||\boldsymbol{r}_k||_2 \leq 10^{-18}\,||\boldsymbol{r}_0||_2 \qquad \text{(for DD)}$$

Table 3 and Figure 5 show the numerical results. In the case of double precision arithmetic, the relative residual norm stagnated at about 1.0×10^{-10} and the solution did not converge. In addtion, the error norm was 9.27×10^0 at 1000 iterations. Because $n = 130$, the residual norm should theoretically converge after 130 iterations. On the other hand, using DD arithmetic that has about twice the number of significant digits, the relative residual norm became 9.89×10^{-19} and the error norm became 2.74×10^{-8} in 18 iterations. This is a great improvement.

It is clear that the reason for the differences is computational error. The error norm did not decrease using double precision arithmetic. Because a double precision floating point number has about 16 (decimal) significant digits, it is difficult to obtain a solution with sufficient accuracy for a system whose condition number is 6.05×10^{10}. In addition, the residual norm stagnates due to computational error when using double precision arithmetic. However, there are situations where the theory-based result for

Table 3. Iteration counts, relative residual norm and relative error norm

	Iteration counts	$\|\boldsymbol{r}\|_2/\|\boldsymbol{r}_0\|_2$	$\|\boldsymbol{x}-\boldsymbol{x}^*\|_\infty/\|\boldsymbol{x}^*\|_\infty$
double	1000	6.28e-11	9.27e+00
DD	18	9.89e-19	2.74e-08

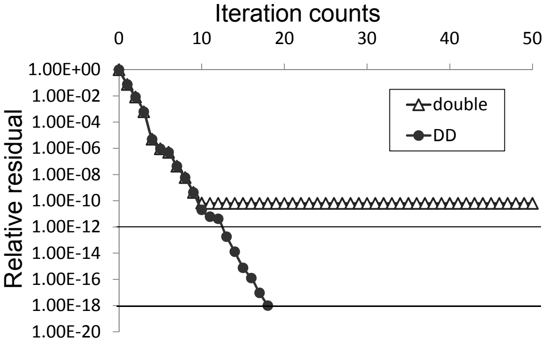


Fig. 5. Convergence history

reducing computational error using DD arithmetic does hold. We also intend to analyze the implementation of these iterative algorithms using QuPAT.

5 Conclusion

To examine certain computational results of double precision arithmetic, we need a higher precision arithmetic environment. For example, in Section 4, in solving a system of linear equations by the GCR(m) method, the relative residual norm may stagnate and may not converge in double precision arithmetic; i.e., it is not possible to obtain the solution with sufficient accuracy. As a useful way to employ quadruple precision arithmetic, Double-Double(DD) is proposed. However, in programming languages such as C, we cannot set up quadruple precision arithmetic easily.

In this paper, we constructed a convenient quadruple precision arithmetic environment QuPAT(Quadruple Precision Arithmetic Toolbox) using Scilab as a toolbox. As a consequence, we were able to use quadruple precision arithmetic without rewriting the code in Scilab, and also utilize mixed precision arithmetic.

As in section 3.1, we defined a new data type ‘dd’ for quadruple precision numbers using the Scilab function ‘tlist’. In QuPAT, a new data type ‘dd’ and the existing data type ‘constant’ were defined separately. Thus it became possible to utilize both double and quadruple precision arithmetic in the same code.

In addition, we applied operator overloading to the type ‘dd’ for the four fundamental rules of DD arithmetic. Thus it became possible to use all of the double, quadruple and mixed precision arithmetic with the same operators $(+, -, *, /)$. The Scilab environment was naturally extended to use DD arithmetic with QuPAT. The computation time for DD arithmetic was about 9 to 15 times that for double precision arithmetic.

QuPAT was implemented using pure Scilab functions. Therefore, if Scilab is available, we can utilize QuPAT independently of underlying hardware and operating systems. QuPAT is downloadable provisionally from the web [9].

Until now, some other important functions, such as sine or cosine were not implemented. In addition, if we use Scilab capability link to functions written in C or Fortran, QuPAT will execute faster. However, if we do use this capability, the code will depends on the computing environments and programming languages. These points will be discussed in future work.

References

[1] Bailey, D.H.: QD (C++ / Fortran-90 double-double and quad-double package), http://crd.lbl.gov/~dhbailey/mpdist/
[2] Dekker, T.J.: A Floating-Point Technique for Extending the Available Precision. Numer. Math. 18, 224–242 (1971)
[3] Hida, Y., Li, X.S., Bailey, D.H.: Quad-double arithmetic: Algorithms, Implementation, and application. Technical Report LBNL-46996, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 (2000)
[4] Hida, Y., Li, X.S., Bailey, D.H.: Algorithms for Quad-Double Precision Floating Point Arithmetic. In: Proceedings of the 15th IEEE Symposium on Computer Arithmetic, pp. 155–162 (2001)
[5] Knuth, D.E.: The Art of Computer Programming, vol. 2. Addison-Wesley, Reading (1969)
[6] Kotakemori, H., Fujii, A., Hasegawa, H., Nishida, A.: Stabilization of Krylov subspace methods using fast quadruple-precision operation. Transactions of JSCES 12, 631–634 (2007) (in Japanese)
[7] MatrixMarket, http://math.nist.gov/MatrixMarket/
[8] Nakasato, N., Ishikawa, T., Makino, J., Yuasa, F.: Fast Quad-Precision Operations On Manycore Accelerators, IPSJ SIG Technical Report (2009) (in Japanese)
[9] QuPAT, http://www.mi.kagu.tus.ac.jp/qupat.html
[10] Scilab, http://www.scilab.org/

Agent Based Evacuation Model with Car-Following Parameters by Means of Cellular Automata

Kohei Arai[1] and Tri Harsono[1,2]

[1] Information Science Department Saga University, Saga Japan
arai@is.saga-u.ac.jp
[2] Electronics Engineering Polytechnic Institute of Surabaya (EEPIS), Surabaya Indonesia
triharsono69@yahoo.com

Abstract. An agent based evacuation model with car-following parameters for micro traffic by means of cellular automata is proposed. The model features smart drivers who have a concern the distance between the driver and the surrounding cars. Such drivers are called agents. Agents lead the other cars not only in an ordinary case but also in the case of evacuations. We put the smart drivers and the agent drivers into the car-following parameters of traffic model. We applied it in the case of evacuation. Experimental simulation results show the effectiveness of reducing the evacuation time by the agents. It also is found the effectiveness increases in accordance with increasing of the number of agents.

Keywords: Agent based traffic model, micro traffic model, car-following, smart driver.

1 Introduction

The traffic flow studies using microscopic simulations (micro traffic model) had been leap occurring with the advancement of computer technology in the last one and half decade [1]…[10]. Neighborhood evacuation plans in an urbanized wild land interface is described by Cova T.J. [11] using agent-based simulation model. They were able to assess spatial effects of a proposed second access road on household evacuation time in a very detailed way. Studies [3]-[7], [10] and [11] enhanced the great benefits of agent-based modeling and simulation in studying emergency evacuation. The effectiveness of simultaneous and staged evacuation strategies using agent-based simulation was presented [12] for three different road network structures. They measured the effectiveness based on total time of evacuation from affected areas.

This study made evacuation simulation of vehicle based on the agent from affected road. The aim is the effectiveness of evacuation speed. It is important to calculate the evacuation speed in the evacuation simulation. It can be information, especially about the evacuation time on disaster affected area in the highway. The next expectation is comparison this simulation results with real data of traffic on a particular area. We conduct micro traffic agent-based modeling and simulation for assessment of evacuation time from the suffered area. In the micro traffic agent-based modeling and

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 71–86, 2010.

simulation, road traffic (probability of vehicle density), driving behavior such as probability of lane changing, car-following are taken into account. The specific parameter in the proposed modeling and simulation is car-following under a consideration of agents. In evacuation condition (evacuation of vehicle), the evacuation speed is the major issue. Therefore, to get the better effectiveness of evacuation speed, we emphasize the improvement of car-following parameter in the traffic model. Smart driver is added to the previously proposed car-following parameter. Although in a state of evacuation, there is possibility that drivers have no panic and can be the smart drivers. It is based on [19] stated that there is irrational behavior during five decades studying scores of disasters such as floods, earthquakes and tornadoes, one of the strongest findings is that people rarely lose control and most survivors who were asked about panic said there was none. In my study the number of smart drivers will be probabilistically determined. A smart driver can be an agent in the typical agent based traffic models. Hereafter, smart drivers are referred to diligent drivers.

We add the parameter of diligent drivers in the car-following with the consideration that the reality on the streets, not all the drivers directly (diligently) to change the speed based on information acquired from the agent or the surrounding areas. When there are some of drivers are not change the speed directly, we call them lazy drivers. By adding diligent drivers in our traffic model, we hope that our system has the sense that it can mimic the basic features of real-life traffic conditions. The diligent drivers determined by probabilistic. Although the proposed simulation is based on the Nagel-Schreckenberg traffic cellular automata [1], lane changing and new car-following parameters are specific to the proposed modeling and simulation.

Following section describes proposed car-following models, then the model used in the proposed micro traffic agent-based modeling and simulation together with the parameter setting for the simulation are followed. After that, implementations of the models as the evacuation system and simulation results are followed by together with some discussions and conclusions.

2 Car-Following Models

In this section, we describe overview of new car-following models. There are two major methods of car-following, continuous and discrete models.

2.1 Continuous Models

In the micro traffic model, the drivers is stimulated their own velocity v_n, the distance between the car and the car ahead s_n, and the velocity of the vehicle in front v_{n+1}. The equation of vehicle motion is characterized by the acceleration function which depends on the input stimuli:

$$\ddot{x}_n(t) = \dot{v}_n(t) = F\left(v_n(t), s_n(t), v_{n-1}(t)\right) \tag{1}$$

Optimal Velocity Model (OVM)

The OVM is a dynamic model of traffic congestion based on a vehicle motion equation. In this model, the optimal velocity function of the headway of the preceding vehicle is introduced. Congestion may occur due to induce by a small perturbation without any specific origin such as a traffic accident or a traffic signal. The OVM can regard to this congestion phenomenon as the instability and the phase transition of a dynamical system [14].

The dynamical equation of the system is obtained (based on the acceleration equation) as,

$$\frac{d^2 x_n(t)}{dt^2} = a\left[V(\Delta x_n(t)) - \frac{dx_n(t)}{dt}\right] \tag{2}$$

where

$$\Delta x_n(t) = x_{n+1}(t) - x_n(t) \tag{3}$$

Which n denotes vehicle number (n = 1, 2… N), N is the total number of vehicles, a state constant representing the driver's sensitivity (which has been assumed to be independent of n), and x_n is the coordinate (location) of the n^{th} vehicle. The OVM assumes that the optimal velocity $V(\Delta x_n)$ of vehicle number n depends on the distance between the vehicle and the preceding vehicle number n+1 (a distance-dependent optimal velocity). When the headway becomes short the velocity must be reduced and become small enough to avoid crash. On the other hand, when the headway becomes longer the driver accelerates under the speed limit, the maximum velocity.

The OVM takes the optimal velocity function $V(\Delta x_n)$ as

$$V(\Delta x_n) = \tanh(\Delta x_n) \tag{4a}$$

$$V(\Delta x_n) = \tanh(\Delta x_n - 2) + \tanh 2 \tag{4b}$$

Eq. (4a) and (4b) respectively are the simple model and the realistic model.

Generalized Force Model (GFM)

The driver behavior is mainly given by the motivation to reach a certain desired velocity v_n (which will be reflected by an acceleration force), and by the motivation to keep a safe distance from other cars (n+1) (which will be described by repulsive interaction forces). The GFM is created to improve the OVM which has the problems of too high acceleration and unrealistic deceleration [15].

In the GFM, one term is added to the right-hand side of equation (2). Thus the formula of the GFM is written by the following equation,

$$\begin{aligned}\frac{d^2 x_n(t)}{dt^2} &= a\left[V(\Delta x_n(t)) - \frac{dx_n(t)}{dt}\right] \\ &+ \lambda\Theta(-\Delta v)(\Delta v)\end{aligned} \tag{5}$$

where Θ denotes the Heaviside function, λ is a sensitivity coefficient different from a. Note that in the GFM, Eq. (5) can be rewritten as follows,

$$\frac{d^2 x_n(t)}{dt^2} = a[v_m - v_n(t)] + a[V(\Delta x_n(t)) - v_m] \\ + \lambda\Theta(-\Delta v)(\Delta v) \tag{6}$$

Note: v_m is the maximum speed. The first term on the right-hand side is the acceleration force, and the last two terms represent the interaction forces.

Full Velocity Difference Model (FVDM)

A full velocity difference model [16] for a car-following theory is based on the previous models. The FVDM model includes car-following parameter to the previously proposed models. Through numerical simulation, property of the model is investigated using both analytic and numerical methods.

On the basis of the GFM formula, taking the positive Δv factor into account, the FVDM is described as the following dynamics equation,

$$\frac{d^2 x_n(t)}{dt^2} = a\left[V(\Delta x_n(t)) - \frac{dx_n(t)}{dt}\right] + \lambda\Delta v \tag{7}$$

The FVDM takes both positive and negative velocity differences into account. The model equation of the FVDM (7) may be reformulated into the following similar form:

$$\frac{d^2 x_n(t)}{dt^2} = a[v_m - v_n(t)] + a[V(\Delta x_n(t)) - v_m] \\ + \lambda\Theta(-\Delta v)\Delta v + \lambda\Theta(\Delta v)\Delta v \tag{8}$$

The GFM assumes that the positive Δv does not contribute to the vehicle interaction, while the FVDM suggests that it does contribute to vehicle interaction by reducing interaction force because $a[V(\Delta x_n(t)) - v_m]$ is always negative and $\lambda\Theta(\Delta v)\Delta v$ is always positive.

Two velocity difference model (TVDM)

TVDM for a car following theory is then proposed taking navigation in modern traffic into account. The property of the model is investigated using linear and nonlinear analysis [17].

Intelligent transportation system (ITS) plays an important role in the rapid development of modern traffic. By using such navigation system, drivers can obtain the information that they need. In accordance with the above concept, on the basis of the OVM, taking both, Δv_n and Δv_{n+1} into account, [17] obtain a more useful model called the two velocity difference model (TVDM), the following dynamics equation is expressed,

$$\frac{d^2 x_n(t)}{dt^2} = a[v_m - v_n(t)] + a[V(\Delta x_n(t)) - v_m] \\ + \lambda\Theta(-\Delta v)\Delta v + \lambda\Theta(\Delta v)\Delta v \tag{9}$$

where $G(.)$ is a generic, monotonically increasing function, and is assumed to be a linear form as,

$$G(\Delta v_n, \Delta v_{n+1}) = r\Delta v_n + (1-r)\Delta v_{n+1} \tag{10}$$

where r is weighting value. In the simulation, selected $p = 0.86$ and known that the influence of the vehicle ahead on the vehicle motion reduces gradually as the distance between the vehicle in concern and the vehicle ahead is increased. Also, the proper value of p could be lead to desirable results.

2.2 Discrete Model: Cellular Automata Model

Cellular automata (CA) are models that are discrete in space, time and state variables. Due to the discreteness, CA is extremely efficient in implementations on a computer. Cellular automata for traffic have been called by traffic cellular automata (TCA).

Stochastic models have been successfully applied to many different interdisciplinary problems. One important example is the modeling of traffic flow using cellular automata. We describe the stochastic model of traffic flow. It is Nagel–Schreckenberg TCA. In 1992, Nagel and Schreckenberg proposed a TCA model that was able to reproduce several characteristics of real-life traffic flows, e.g., the spontaneous emergence of traffic jams [1]. Their model is called the *NaSch TCA*, but is more commonly known as the *stochastic traffic cellular automaton* (STCA). The road is divided into sections of a certain length Δx and the time is in discrete to steps of Δt. Each road section can either be occupied by a vehicle or empty and the dynamics are given by update rules of the form velocity $v_n(t+1)$ and distance $x_n(t+1)$.

The computational model in the STCA is defined on a one-dimensional array of L sites and with open or periodic boundary conditions. Each site may either be occupied by one vehicle or it may be empty. Each vehicle has an integer velocity with value between zero to $v_{\max}$ (maximum speed). For an arbitrary configuration, one update of the system consists of the following four consecutive steps, which are performed in parallel for all vehicles:

1) Acceleration

$$\begin{gathered} v_{(i,j)}(t\text{-}1) < v_{max} \wedge gs_{(i,j)}(t\text{-}1) > v_{(i,j)}(t\text{-}1) + 1 \\ \Rightarrow v_{(i,j)}(t) \leftarrow v_{(i,j)}(t\text{-}1) + 1 \end{gathered} \tag{11}$$

($gs_{(i,j)}(t)$ is space gap at each time step t for vehicle in the i^{th} lane and j^{th} position or the distance to the next vehicle ahead; $v_{(i,j)}(t)$ is vehicle speed for vehicle in the i^{th} lane and j^{th} position).

2) Braking

$$gs_{(i,j)}(t\text{-}1) \le v_{(i,j)}(t\text{-}1) \Rightarrow v_{(i,j)}(t) \leftarrow gs_{(i,j)}(t\text{-}1) - 1 \tag{12}$$

3) Randomization

$$\xi(t) < p \Rightarrow v_{(i,j)}(t) \leftarrow \max[0, v_{(i,j)}(t) - 1] \tag{13}$$

($\xi(t)$ is random number, p is stochastic noise parameter or slowdown probability).

4) Vehicle movement

$$x_{(i,j)}(t) \leftarrow x_{(i,j)}(t - 1) + v_{(i,j)}(t) \tag{14}$$

Through the step one to four very general properties of single lane traffic are modelled on the basis of integer valued probabilistic cellular automaton rules. Already this simple model shows nontrivial and realistic behavior. Step 3 is essential in simulating realistic traffic flow otherwise the dynamics is completely deterministic. It takes into account natural velocity fluctuations due to human behavior or due to varying external conditions. Without this randomness, every initial configuration of vehicles and corresponding velocities reaches very quickly at a stationary pattern which is shifted backwards (i.e. opposite the vehicle motion) in one site per time step.

3 Proposed Car-Following Model

The particular challenge of the proposed vehicle dynamics model is to express the drivers' characteristic dependence on the factors such as perceptions, psychological motivations and reactions, or social behaviors. Thus, in contrast to physical processes, drivers' behavior cannot be expected to be descriptive by a few natural variables [15]. In this study, we proposed one of the driver behaviors is about smart driver (diligent driver). The number of diligent driver is probabilistically determined. The characteristic of the diligent drivers is that they have a concern about the distance between their vehicle and the vehicle in front so that they will change the speed based on the aforementioned conditions. The proposed micro traffic model is based on the agent model. A diligent driver becomes an agent driver by fulfills certain characteristics. Both the diligent driver or the agent driver have the same characteristic, it is about the additional speed. The difference between both is a diligent driver has additional speed $c = [0 : \min(\bar{v}, v)]$ while an agent has $c' = [0 : \max(v)]$.

Through a consideration of road of length L, with N vehicles, n^{th} vehicle length s_0 and a distance between current vehicle to the next vehicle ahead s_n; $n = 1\ldots N$ as is shown in Fig. 1. The distance between the current vehicle and the next vehicle ahead is determined randomly because of $s_n \neq s_{n+1}$. The car density k is given by:

$$k = \frac{N}{L} = \frac{N}{\left(Ns_0 + \sum_{n=1}^{N} s_n\right)} \tag{15}$$

So that the car flow rate q in this study is given by:

$$q(k) = vk = \frac{Nv}{\left(Ns_0 + \sum_{n=1}^{N} s_n\right)} \tag{16}$$

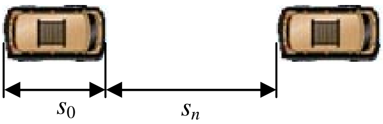


Fig. 1. n^{th} vehicle length is s_0 and the distance between the current vehicle and the next vehicle ahead of the current car s_n

The velocity v of each vehicle is given by the headway to the vehicle in front. In the proposed model, the velocity of the diligent driver is different from the velocity of the agent car. The velocity v is given by:

$$v = \begin{cases} v_{(i,j)}(t) + c = v_{(i,j)}(t) + [0 : \min(\bar{v}, v)] \\ for \quad diligent \quad driver \\ \\ v_{(i,j)}(t) + c' = v_{(i,j)}(t) + [0 : \max(v)] \\ for \quad agent \quad driver \end{cases} \tag{17}$$

We make initial condition (arrival distribution) of vehicle by use normal distribution. It need mean speed of vehicle $\bar{v}$ to create initial speed of vehicles. We also define parameter c in the speed of diligent driver is a function of mean speed $\bar{v}$. To get an intuitive feeling for proposed system dynamics, the two time–space diagrams are shown in Fig.2. Both diagram shows the evolution for a global density of k = 0.2 vehicles/cell., but with diligent driver (*dd*) set to 0.25 for the diagram (a), and *dd* = 0.5 for the diagram (b). In both diagram of those, the randomization in the model gives increasing many unstable *artificial phantom mini-jams* (phantom mini-jams is traffic jam can sometimes form for no apparent reason). Black area in Fig.2 showed the traffic jam. The downstream fronts of these jams smear out, forming unstable interfaces. This is a direct result of the fact that the intrinsic noise (as embodied by *p*) in the STCA model is too strong: a jam can always form at any density, meaning that breakdown may (and will) occur, even in the free-flow traffic regime [13].

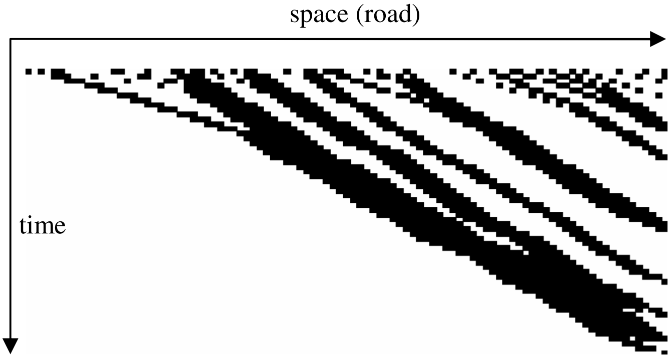
For low enough densities, however, these jams can be vanished as they are absorbed by vehicles with sufficient space headways or by new jams in the system [13]. Being stated by [18] that when vehicles are not impeded by other traffic they travel at a maximum speed (*free speed*) then at free speed, flow rate *q* and density *k* will be close to zero.

k-*q* diagram is shown in Fig.3 as a simulation results with the proposed model. At saturated roads (large density), the flow rate *q* is saturated (the vehicles are queuing). This situation is appropriate with phenomena of traffic flow stated by [18][13].

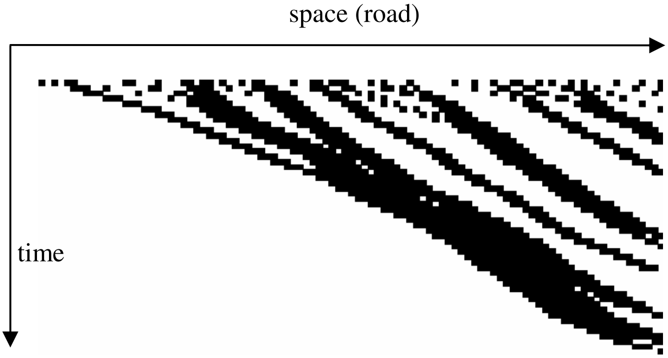
Our proposed car-following model is given as,
A smart driver (*diligent driver*):

$$x(t) = x(t-1) + v(t) + c \tag{18}$$

with $gs(t\text{-}1) > v(t - 1)$ and $c = [0 : \min(\bar{v}, v)]$.



(a) At a density 0.2 (20%) and diligent driver 0.25 (25%)



(b) At a density 0.2 (20%) and diligent driver 0.5 (50%)

Fig. 2. Simulated traffic in time-space domain. Each new line shows the traffic lane after one further complete velocity-update and just before the car motion. A single lane is assumed.

An agent:

$$x(t) = x(t-1) + v(t) + c' \tag{19}$$

with $gs(t\text{-}1) > v(t - 1)$ and $c' = [0 : \max(v)]$.

The proposed car-following based on the STCA model [1], there is specific difference between both of them. CF of the STCA model Eq. (14) has foregoing position of car $x(t - 1)$ and the current speed $v(t)$ (on the right hand side) while the proposed CF, either CF for a diligent driver Eq. (18) or CF for an agent Eq.(19) has the additional parameter added into the right hand side, it is about additional speed.

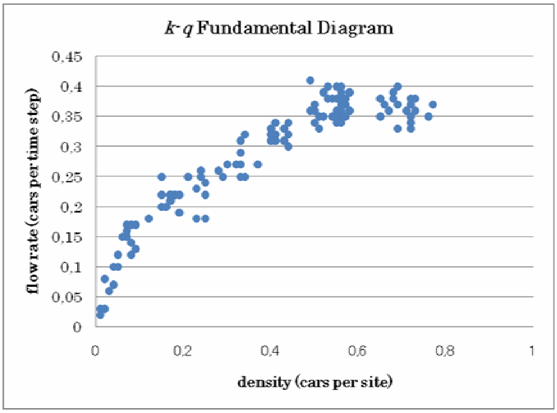


Fig. 3. Flow rate *q* (cars per time step) *vs* density *k* (cars per site) from simulation results (*k-q* diagram)

4 Extended the NaSch Cellular Automata Model

The STCA Eq. (11) to Eq. (14), is a minimal model in the sense that all these rules are a necessity for mimicking the basic features of real-life traffic flows [13]. In the proposed model, lane changing and car-following parameters are added into the STCA model. Two lanes of traffic (multi-lane traffic) are taken into account in the simulation and observation.

The proposed lane changing has rules and sub steps:

1) lane changing (2 lanes)

$$\begin{gathered} gs_{(i=1,\,j)}(t) < v \wedge x_{(i=2,\,j,\,j+v)}(t)=0 \\ \Rightarrow x_{(i=2,\,j+a)}(t) \leftarrow x_{(i=1,\,j)}(t\text{-}1) \end{gathered} \tag{20}$$

with probability prob. of lane changing *PL* and $a = [0 : v]$

or

$$\begin{gathered} gs_{(i=2,\,j)}(t) < v \wedge x_{(i=1,\,j,\,j+v)}(t)=0 \\ \Rightarrow x_{(i=1,\,j+a)}(t) \leftarrow x_{(i=2,\,j)}(t\text{-}1) \end{gathered} \tag{21}$$

with probability prob. of lane changing *PL* and $a = [0 : v]$.

2) car-following

$x_{(i,\,j)}(t)$ a smart driver (diligent driver):

$$x(t) = x(t-1) + v(t) + c \tag{22}$$

with $gs(t\text{-}1) > v(t\text{ - }1)$ and $c = \left[0 : \min(\bar{v}, v)\right]$.

$x_{(i,\,j)}(t)$ an agent:

$$x(t) = x(t-1) + v(t) + c' \tag{23}$$

with $gs(t\text{-}1) > v(t - 1)$ and $c' = [0 : \max(v)]$.

We get the complete rules of the proposed system by adding lane changing and car-following (Eq. (20)...(23)) into the STCA model (Eq. (11)...(14)):

1) *acceleration*

$$\begin{gathered} v_{(i,\,j)}(t\text{-}1) < v_{\max} \wedge gs_{(i,\,j)}(t\text{-}1) > v_{(i,\,j)}(t\text{-}1) + 1 \\ \Rightarrow v_{(i,\,j)}(t) \leftarrow v_{(i,\,j)}(t\text{-}1) + 1 \end{gathered} \tag{24}$$

2) *braking*

$$\begin{gathered} gs_{(i,j)}(t\text{-}1) \leq v_{(i,\,j)}(t\text{-}1) \\ \Rightarrow v_{(i,\,j)}(t) \leftarrow gs_{(i,j)}(t\text{-}1) - 1 \end{gathered} \tag{25}$$

3) *randomization*

$$\xi(t) < p \Rightarrow v_{(i,\,j)}(t) \leftarrow \max[0,\, v_{(i,\,j)}(t) - 1] \tag{26}$$

4) *car-following*

$x_{(i,\,j)}(t)$ a smart driver (diligent driver) :

$$x_{(i,\,j)}(t) \leftarrow x_{(i,\,j)}(t - 1) + v_{(i,\,j)}(t) + c \tag{27}$$

with $gs(t\text{-}1) > v(t\text{-}1)$ and $c = [0 : \min(\bar{v}, v)]$.

$x_{(i,\,j)}(t)$ an agent :

$$x_{(i,\,j)}(t) \leftarrow x_{(i,\,j)}(t - 1) + v_{(i,\,j)}(t) + c' \tag{28}$$

with $gs(t\text{-}1) > v(t\text{-}1)$ and $c' = [0 : \max(v)]$.

5) *lane changing*

$$\begin{gathered} gs_{(i=1,\,j)}(t) < v \wedge x_{(i=2,\,j,\,j+v)}(t)=0 \\ \Rightarrow x_{(i=2,\,j+a)}(t) \leftarrow x_{(i=1,\,j)}(t\text{-}1) \end{gathered} \tag{29}$$

with probability prob. of lane changing PL and $a = [0 : v]$.

or

$$\begin{gathered} gs_{(i=2,\,j)}(t) < v \wedge x_{(i=1,\,j,\,j+v)}(t)=0 \\ \Rightarrow x_{(i=1,\,j+a)}(t) \leftarrow x_{(i=2,\,j)}(t\text{-}1) \end{gathered} \tag{30}$$

with probability prob. of lane changing PL and $a = [0 : v]$.

The rule above for increasing the speed of a vehicle and braking to avoid collision, i.e., rule Eq. (24) and Eq. (25) as well as rule Eq. (27) and Eq. (28) for the actual vehicle movement. Eq. (26) is stochastic in the system. At each time step t, a random number $\xi(t) \in [0,1]$ is drawn from a uniformly distributed random number. This number is then compared with a stochastic noise parameter $p \in [0,1]$ called the slowdown probability. As a result, there is a probability of p that a vehicle will slow down to $v_{(i,j)}(t) - 1$ cells/time step. According to the Nagel and Schreckenberg [1], the randomization of rule Eq. (26) captures natural speed fluctuations due to human behavior or varying external conditions. The rule introduces overreactions of drivers' behavior when braking which provides the key to the formation of spontaneously emerging jams.

5 Implementation in the Evacuation System

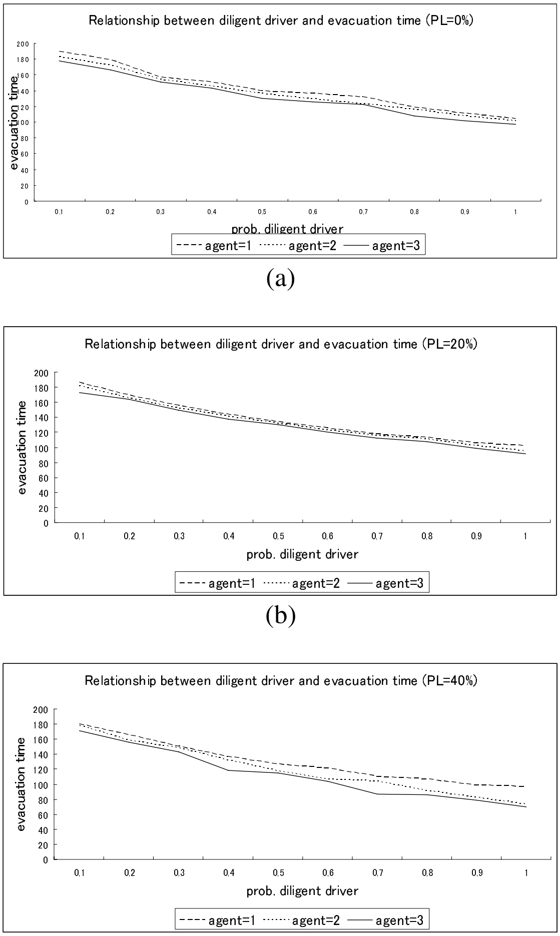
In the evacuation system, we apply our proposed car-following approach on micro traffic with the specification: road shape straight road, road length 200 cells, 2 traffic lanes, mean speed of vehicle 3 cells/time steps, and deviation of vehicle speed 2 cells/time steps. Evacuation time is measured by the total time needed for evacuation. We observed relation between the smart driver (diligent driver) and evacuation time, likewise relationship the number of agent with respect to evacuation time. Besides that it also evaluated the relationship between vehicle speed (mean speed) and evacuation time. The effectiveness of the agent is shown in terms of evacuation time.

Fig.4 show relation among the probability of diligent driver *dd* and the evacuation time *T* on the different number of agents (agent=1 the dashed line, agent=2 the dotted line, and agent=3 the straight line). We showed some of experiment results in Fig.4a, 4b, and 4c. The first result (Fig. 4a) using the density k = 60% and the probability of lane changing PL = 0%, the second one (Fig. 4b) k = 60% and PL = 20%, and the third one (Fig. 4c) k = 60% and PL = 40%. Either first result of simulation, second one or third one (with different PL) has the same pattern, when the probability of diligent driver *dd* increases causes the evacuation time *T* decrease (its mean is the evacuation speed greater). It occurs in all different number of agents. For instance in the first result of simulation using k = 60%, PL = 20% (Fig.4b), at the *dd* = 60% obtained the evacuation time T = 126, 123, and 120 consecutively for the number of agent = 1, 2, and 3. From the results it can be said that by the increasing of the number of agents cause the evacuation time decreases.

The third result of the simulation in the Fig.4b (k = 60% and PL = 40%). We take one example of the simulation result using the probability of diligent driver *dd* = 80%, we obtained the evacuation time T = 107, 91, and 86 consecutively for the number of agent = 1, 2, and 3. From the results it also can be said that by the increasing number of agents cause evacuation time decreases. The different values *dd* also showed that the evacuation time *T* decreases when the number of agents increases.

Of course it can be stated that the evacuation time decreases when the number of agents increase. Overall simulation results can be stated that the increasing of the number of agents and the number of diligent drivers (in probability) cause the decreasing of the evacuation time in the micro traffic.

In the simulation we also observed the evacuation time *T* with respect to the mean speed $\overline{v}$. The mean speed $\overline{v}$ is a parameter of diligent driver in the car-following driving behavior. We described the complete dynamics of the vehicle's positions x for the diligent driver in the Eq. (18). It can be rewritten by



(a)

(b)

(c)

Fig. 4. Relationship between the probability of diligent driver *dd* and the evacuation time *T*, it is done on the different number of agents. (4a): using the density k = 60% and probability of lane changing PL = 0; (4b): k = 60% and PL =20%; (4c): k = 60% and PL =40%.

$$\begin{gathered} x_{(i,j)}(t) = x_{(i,j)}(t-1) + v_{(i,j)}(t) \\ +\left[0 : \min(\overline{v}, v)\right] \end{gathered} \tag{31}$$

We found relation between the number of agent and evacuation time (Fig.5). It can be said that the evacuation time will decrease when we add the number of agent in the evacuation simulation. For instance using probability of diligent driver *dd* = 70%,we obtained evacuation time 110, 104, 87 respectively for agent = 1, agent = 2, and agent = 3 (the straight line, Fig.5). We also see that using different probability of diligent driver *dd* produced different evacuation time. The evacuation time will decrease when the value of *dd* greater. For instance by use *dd* = 70% (the straight line, Fig.5) produce evacuation speed larger than by use *dd* = 50% (the dotted line, Fig.5). On the other hand we can say that using the number of agent greater, we will have the evacuation speed greater too.

Fig.6 is simulation results using density *k* = 60%, probability of lane changing *PL* = 40%, and probability of diligent driver *dd* = 80%. The simulation results showed that the evacuation time *T* decreases when the number of agent increase in the certain mean speed $\overline{v}$. This situation occurs for all kind of the number of agents, either the number of agent 1, 2 or 3. For example using mean speed $\overline{v} = 2$ we obtained the evacuation time *T* = 111, 108, and 104 consecutively for the number of agents 1, 2, and 3. Otherwise, in the certain number of agent we observed that the evacuation time *T* decreases by increasing the mean speed $\overline{v}$. Overall simulation we can said that by increasing the mean speed $\overline{v}$ and the number of agents cause the evacuation time *T* decreases.

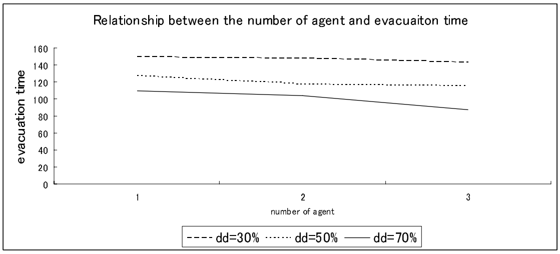
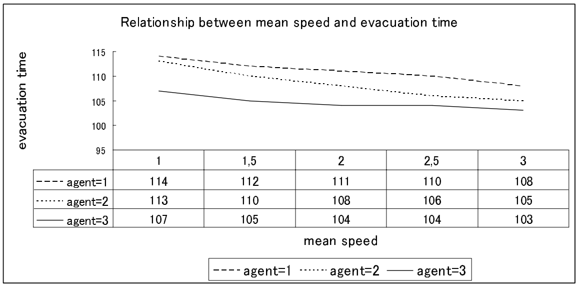


Fig. 5. Relationship between the number of agent and the evacuation time, the dashed line using prob. diligent driver *dd* = 30%, the dotted line using *dd* = 50%, and the straight line using *dd* = 70% (all use density *k* = 60% and prob. lane changing *PL* = 40%)



	1	1,5	2	2,5	3
agent=1	114	112	111	110	108
agent=2	113	110	108	106	105
agent=3	107	105	104	104	103

Fig. 6. Relationship between the mean speed $\overline{v}$ and the evacuation time *T* on the different number of agents using density *k* = 60%, probability of lane changing *PL* = 40%, and probability of diligent driver *dd* = 80%

The comparison between previous model (NaSch cellular automata/STCA) and our proposed model has been done. Fig.7 show the evacuation time *T* with respect to the probability of diligent driver *dd*. Based on the simulation result we can say that using the previous model has evacuation time larger than using the proposed model in the

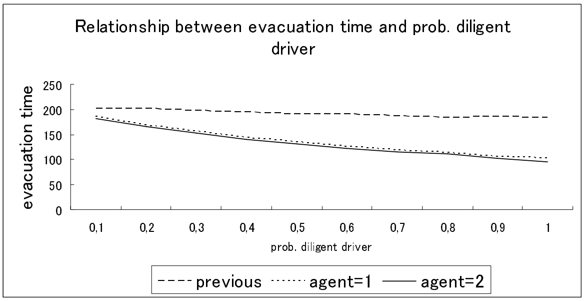
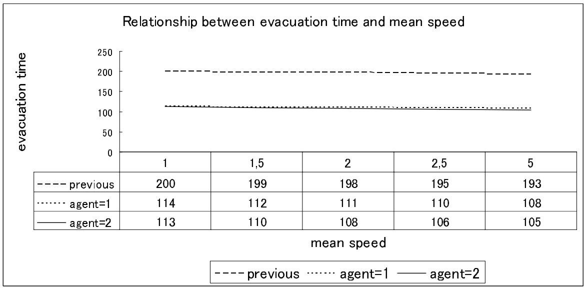


Fig. 7. Comparison among previous model and proposed model, the dashed line is previous model, the dotted line and the straight line is proposed model using 1 and 2 agents respectively



	1	1,5	2	2,5	5
---- previous	200	199	198	195	193
······· agent=1	114	112	111	110	108
—— agent=2	113	110	108	106	105

Fig. 8. Comparison among previous model and proposed model, the dashed line is previous model, the dotted line and the straight line is proposed model using 1 and 2 agents respectively

same value of diligent driver *dd*. For instance using *dd* = 90% we have comparison result of evacuation time i.e. T = 185 (previous model), T = 106 and T = 102 (for number of agent = 1 and 2 respectively in the proposed model). Thus by use the proposed model (agent based model) we obtain evacuation speed larger than by use the previous model.

We also make a comparison of evacuation time with respect to the mean speed between the previous model and the proposed model. The evacuation simulation results in Fig.8. We can see in the table below of the graphics (Fig.8), in the every mean speed the value of evacuation time in the previous model (the dashed line) larger than the value of evacuation time in the proposed model (using agent = 1 and 2, the dotted line and the straight line respectively). For instance using mean speed 2 cell/time step, we have evacuation time T = 198 (previous model), T = 111 and T = 108 (for number of agent = 1 and 2 respectively in the proposed model). We can say that using the proposed model, we obtain the evacuation time more quickly than using the previous model.

The effectiveness of the agent has been observed in terms of evacuation speed (evacuation time). Table 1 show the comparison results between the previous model and the proposed model about the evacuation time T with respect to the probability of diligent driver *dd*. We have the effectiveness of the agent and in the Table 1 we show

Table 1. The effectiveness of the agent in terms of evacuation time

dd (%)	Evacuation time				
	Previous model	**Prop. (*A*=1)**	**Effectiveness (%)**	**Prop. (*A*=3)**	**Effectiveness (%)**
10	202	186	8	173	14
20	201	169	16	164	18
30	197	156	21	149	24
40	195	144	26	138	29
50	191	134	30	130	32
60	190	126	34	120	37
70	187	118	37	112	40
80	184	113	39	108	41
90	185	106	43	99	46
100	183	102	44	91	50

Note: Prop. = proposed model and *A* = the number of agent.

one example of the results using 1 agent and 3 agents. The effectiveness of the agent can be seen that using 1 agent we obtain the effectiveness value more increase by the increasing of diligent driver *dd*. This condition also occurs when we use 3 agents.

6 Concluding Remarks

In spite of the Nagel Schreckenberg model does not take meta stability into account, the comparative simulation study between with and without agent as well as diligent drivers is conducted based on the Nagel Schreckenberg model. Namely the purpose of this paper is to show how effective such drivers to reduce evacuation time as well as reducing the number of victims. The simulation results show that the effect of diligent driver depends on the percentage ratio of diligent driver and is almost double when the percentage ratio of diligent driver is 100%. It is found that the effect of agent driver also depends of the number of agent driver and is almost double when the number of agent driver is three in comparison to the existing simulation result without any agent. Also it is found that the evacuation time is reduced 12% when the lane change is permitted (when we compare the evacuation times by using the probability of lane change is set at 0.2 and 0.4).

References

1. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. Journal Physics I, France 2, 2221–2229 (1992)
2. Nagel, K.: Particle hopping models and traffic flow theory. Phys. Rev. E 53, 4655–4672 (1996)

3. Sugiman, T., Misumi, J.: Development of a new evacuation method for emergencies: Control of collective behavior by emergent small groups. J. Appl. Psychol. 73, 3–10 (1988)
4. Stern, E., Sinuany-Stern, Z.: A behavioral-based simulation model for urban evacuation. Pap. Reg. Sci. Assoc. 66, 87–103 (1989)
5. Sheffi, Y., Mahmassani, H., Powell, W.: A transportation network evacuation model. Transport Res. A-Pol. 16, 209–218 (1982)
6. Hobeika, A.G., Jamei, B.: MASSVAC: A model for calculating evacuation times under natural disaster. Emerg. Plann, Simulation Series 15, 23–28 (1985)
7. Cova, T.J., Church, R.L.: Modeling community evacuation vulnerability using GIS. International Journal Geographic Information Science 11, 763–784 (1997)
8. Deadman, P.J.: Modeling individual behavior and group performance in an intelligent agent-based simulation of the tragedy of the commons. J. Environ. Manage. 56, 159–172 (1999)
9. Teodorovic, D.A.: Transport modeling by multi-agent systems: A swarm intelligence approach. Transport Plan. Techn. 26, 289–312 (2003)
10. Church, R.L., Sexton, R.M.: Modeling small area evacuation: Can existing transportation infrastructure impede public safety? Caltrans Testbed Center for Interoperability Task Order 3021Report, Vehicle Intelligence & Transportation Analysis Laboratory, University of California, Santa Barbara
11. Cova, T.J., Johnson, J.P.: Micro simulation of neighborhood evacuations in the urban-wild land interface. Environ. Plann. A 34, 2211–2229 (2002)
12. Chen, X., Zhan, F.B.: Agent-based modeling and simulation of urban evacuation: relative effectiveness of simultaneous and staged evacuation strategies. Journal of the Operational Research Society 59, 25–33 (2008)
13. Maerivoet, S., De Moor, B.: Cellular automata models of road traffic. Physics Reports 419, 1–64 (2005)
14. Bando, M., Hasebe, K., Nakayama, A., Shibata, A., Sugiyama, Y.: Dynamical model of traffic congestion and numerical simulation. Phys. Rev. E 51, 1035–1042 (1995)
15. Helbing, D., Tilch, B.: Generalized force model of traffic dynamics. Phys. Rev. E 58, 133–138 (1998)
16. Jiang, R., Wu, Q.S., Zhu, Z.J.: Full velocity difference model for a car-following theory. Phys. Rev. E 64, 017101-1–017101-4 (2001)
17. Ge, X.H., Cheng, R.J., Li, Z.P.: Two velocity difference model for a car following theory. Physica A 387, 5239–5245 (2008)
18. Immers, L.H., Logghe, S.: Traffic Flow Theory, Faculty of Engineering, Department of Civil Engineering, Section Traffic and Infrastructure. In: Kasteelpark Arenberg, B-3001 Heverlee, Belgium, vol. 40 (2002)
19. Kretz, T., Mosbach: Pedestrian Traffic, Simulation and Experiment, Doctor Dissertation, Vom Fachbereich Physik der Duisburg-Essen University (2007)

A Cellular Automata Based Approach for Prediction of Hot Mudflow Disaster Area

Kohei Arai and Achmad Basuki

Dept. of Information Science, Saga University
Saga, Japan
arai@is.saga-u.ac.jp, basukieepis2008@yahoo.com

Abstract. A Novel cellular automata's based approach for prediction of hot mudflow disaster is proposed. A prediction model for hot mudflow based on fluid dynamic is proposed because hot mudflow spread like fluid dynamic with velocity, viscosity and thermal flow parameters. We use much simpler cellular automata's approach with adding some probabilistic parameter because that is relatively simple and have a good enough performance for visualization of fluid dynamics. We add some new rules to represent hot mudflow movement such as moving rule, precipitation rule, and absorption rule.

The prediction results show high accuracy of elevation changes at the predicted points and its surrounding areas. We compare these predicted results to the digital elevation map derived from ASTER/DEM. Some period maps to evaluate the prediction accuracy of the proposed method.

Keywords: hot mudflow, prediction model, cellular automata, Gaussian function, fluid dynamics.

1 Introduction

Sidoarjo hot mudflow disaster is one of the biggest unstoppable disasters that occurred on May 29th 2006 suddenly caused by a gas exploration. During the first three years, the disaster destroyed some villages, thousand of houses and buildings, farming, schools, markets and factories. The weight of the mud on the ground was reported already and corresponding to the weight for pressing down a large area of Sidoarjo land by approximately one meter. Nowadays mud blows around 100,000m^3 per day [1]. It is also reported that the plumed mud contains 70% of water. It implies that 687,000-barrel water spread out every day. How big impact of disaster are in environment, economic and human resource in the future when this disaster cannot be stopped [2].

Assuming that mudflow is similar to fluid flow, a fluid-flow model creates the prediction model of mudflow movement. The simple fluid flow model proposed by Argentini [3] uses Cellular Automata that are proposed here. This model is useful for visualization of fluid flow phenomena with some parameters such as volume, velocity and obstacle avoidance. This model cannot be used for mudflow simulation because it does not handle viscosity and thermal parameter. The other model is lava flow model

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 87–98, 2010.

proposed by Vicari [4] that is based on a Cellular Automata approach. This model is better model for representation of hot mudflow because it can treat the parameters, volume, velocity, viscosity and thermal situations.

The Cellular Automata approach can visualize hot mudflow disaster in free-space area. It, however, is necessary to add some additional approaches for visualization of the actual conditions those are not only natural conditions, but also human factor parameters such as dike, building and road. Thus a combination of Argentini's and Vicari's models is proposed. It should be a better prediction model of hot mudflow spreading with a consideration of the human factors. Due to the fact that the Argentini's model uses limited integer state and Vicari's model uses floating point state, the proposed model uses a discrete model with floating points.

On precursor model, we can visualize mudflow movement like a combination between fluid flow and lava flow models. Although the model has a good visualization capability of mudflow movement, we have to add some properties, map data and rules to make it better to show the actual conditions that have some obstacles like dikes and building. The proposed prediction model is to inform of where some inundated locations are. It will be used to restrain geological impact of hot mudflow disaster. We use some basic parameters of dynamic system to simulate prediction model. That is an approximation model to describe actual model for representation of actual situations.

We use ASTER (Advanced Spaceborne Thermal Emission and Reflection)/DEM (Digital Elevation Model) data for landscape map of the disaster area and its surrounding area on some periods in order to show how the prediction results and the actual situations differ. Prediction accuracy of the proposed method is also compared to some other fluid-flow models with the reference to the ASTER/DEM derived elevation as a true landscape map.

2 Theoretical Background

2.1 Figures

Cellular automata (CA) are a set of array of automata called cell. They interact one to another cells. Array model of CA is expressed with one-dimensional shape, two-dimensional (2D) model grids, or three-dimensional (3D) solids as are shown in Figure 3. Almost all the cells are aligned at the simple lattice points, but are aligned in a complicated form like honeycomb in other rules. Finally, CA are simple model to describe the complex system of life.

As simple model, Cellular Automata only have three fundamental properties, state, neighborhood and program. The state is a given variable for defining each cell. It can be shown in numbers or properties. In simple way, each cell is written as sub-landscape; therefore state is sum of *individual* location or type of growing area. The neighborhood is a set of cells. That interact each other in the physical grid, and two fundamental neighborhood models are Von-Nuemann Neighborhood and Moore Neighborhood as are shown in Fig. 1. The program is a set of defined rules to change state as response in a time depending on its neighborhood. In CA model approach, we can develop some new rules based on state condition and neighborhood.

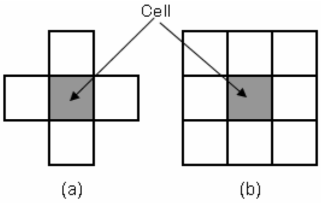


Fig. 1. Neighbor model in 2D Cellular Automata (a) The Von Neumann Neighborhood (b) The Moore Neighborhood

Mathematical point of view of cellular automata, the state at position (*x,y*) at time t that is written $s_t(x,y)$ will change into $s_{t+1}(x,y)$ at time *t+1* with rule that can be written:

$$s_{t+1}(x,y) = \underset{u,v}{\otimes} s_t(x+\delta_x, y+\delta_y) \tag{1}$$

Where δ_x, δ_y are position of its neighbors.

Formally cellular automata have three basic components [12] such as:

- A regular lattice of cells covering a portion of n-dimensional space.
- A set of $s_t(x,y)=\{s_t^{(1)}(x,y), s_1^{(2)}(x,y), \ldots, s_t^{(m)}(x,y)\}$ of Boolean variables attached to site (x,y) of the lattice and giving a local state of each cell at time t=1,2,3,….
- A rule $R=\{R_1, R_2, \ldots, R_k\}$ which specific function of state $s_t(x,y)$ that is written:
  $\mathbf{s}_{t+\tau}^{(j)}(x,y)=R_j\{s_t^{(j)}(x,y), s_t^{(j)}(x+\delta x_1, y+\delta y_1), \ldots s_t^{(j)}(x+\delta x_m, y+\delta y_m)\}$

Where $(x+\delta x_j, y+\delta y_j)$ is given neighbor.

2.2 Fluid Dynamic Celullar Automata

CA model can be used to describe fluid dynamic phenomena such as fluid flow, lava flow and gases dynamic. There are four types of update state changing model in fluid dynamic CA as shown in Fig. 2, such as growth model [4][7], Icing-Like dynamic model [7], moving model [3][8] and majority model [9]. These models have different state types and update state rules. Many fluid-dynamic models use growth model and moving model such as Argentini's model and sea-wave simulation. The other models - Icing-Like model and majority model – usually use to append the properties of fluid dynamic such as viscosity as shown as lava flow model and mudflow model.

Displayed equations or formulas are centered and set on a separate line (with an extra-line or half-line space above and below). Displayed expressions should be numbered for reference. The numbers should be consecutive within each section or within the contribution, with numbers enclosed in parentheses and set on the right margin.

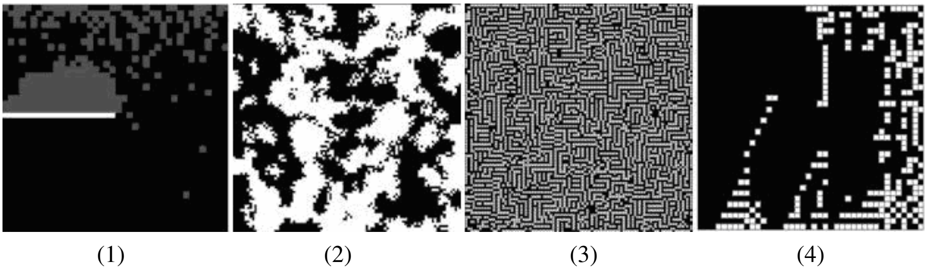


Fig. 2. Four update state changing models in cellular automata (1) Growth model (2) Majority model (3) Icing-like model (4) Moving model

The main focus in fluid dynamic is update rules, and the main processes in this rule are collision between cells, and cell moving. The example common rules in fluid dynamic, introduced by Albertini [3] as shown in Fig. 3, is the rule of cell moving, because this rule run if the state have fluid particle or $s_t(i,j)$=1. These rules use Moore neighborhood, and the direction of cell moving depends on neighbor position because it relates on force of particle interaction. These rules also are basic simple rule of fluid dynamic for simulate flood hazard [8].

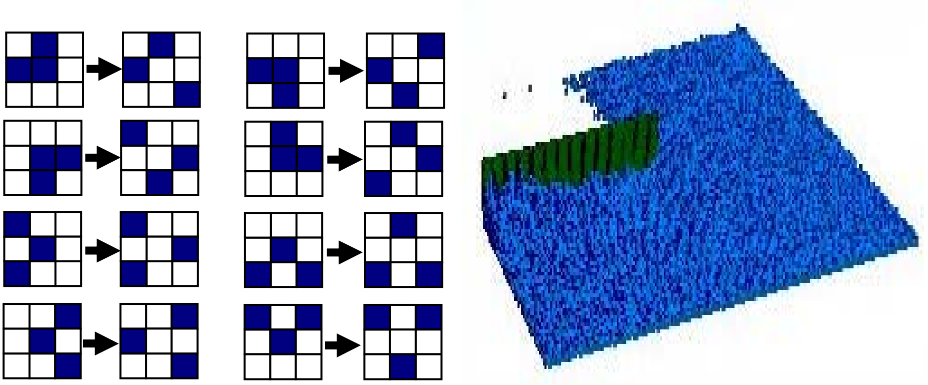


Fig. 3. Example rule and result in Albertini's Model [3]

Another model for fluid flow model, developed by Avolio [10], is Cellular Automata model for simulation of 1992 Tessina Landslide. This model is mudflow model, and used Von-Neumann neighborhood. It is quite different with Argentini's model because this model uses floating point states. This model is simple and useful, and has good performance for landslide caused by mudflow. But on high volume mud blows, this model has a problem to identify how much mud will move to other area because there is no eliminated cell on center of mud blows.

Because of that condition, we propose combination of basic rule on Albertini's model and Vicari's model that makes a new Cellular Automata approach to simulate hot mud flow that has a good performance to predict where mud will flow in the future.

2.3 Minimization Rules

Fig. 4 shows the CA rules that are used on some fluid flow simulations. These rules are called as Minimization Rules. The minimization rules describe how the material will flow from a cell to its neighbors. D'Ambrosio [11] shows the description of minimization rule for soil erosion by water. In these rules we have two values E(i) as number of solid material and H(i) as number of fluid material:

a. E(1)>H(0), E(2)>H(0) → cell 1 and 3 eliminated
b. Er=31/3=10.3, E(4)<Er → cell 4 eliminated
c. Er=20/2=10, no cell eliminated
d. Reconfiguration □H=Er-min(E)=(10-6)=4 → E(0)=7; E(2)=10

Although these rules show the material transport, but it cannot show how much materials will move and how much materials will stay on a cell because it will change from fluid to soil.

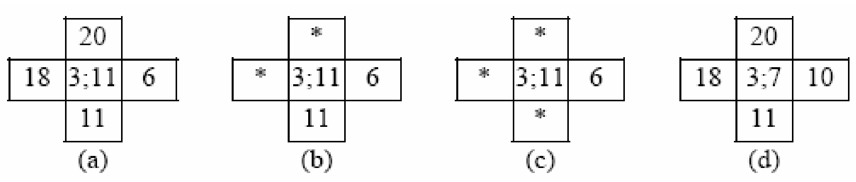


Fig. 4. Minimization rules by D'Ambrosio for soil erosion by water[10]

3 Proposed Method

Hot mudflow model is similar to fluid dynamic flow model. We, however, need to combine all basic updating state models of CA to make it look like real condition because the state properties in each algorithm is binary state unless growth model, otherwise mudflow model is floating point model. In this research, we combine fluid dynamic flow model from Argentini and lava flow model. Our model uses Argentini's model as primary model because the Argentini's model is very simple model based on growth model to describe fluid dynamic flow with discrete state. The Argentini's model, however, is not enough to describe mudflow because this model does not show some fluid particles such as viscosity, erosion and deposition.

3.1 Variables

As many fluid-dynamic Cellular Automata models, our model uses 2D Cellular Automata with Moore neighborhood (8 neighbor nodes). We use floating-point states in order to describe current states of mudflow map, and those are different with Albertini's model because floating-point state is easier to define map data similar with real data. The state S is float between 0 and 1. In this research, we define three-type variables: mud $s_t(x,y)$, ground $h_t(x,y)$ and dike (use same variable with ground because the dike have same characteristics with ground), as shown in fig. 5.

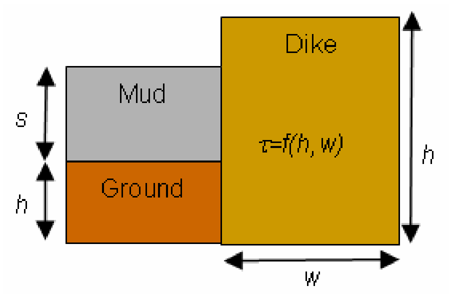


Fig. 5. Three types variables

3.2 Rules

We define *s(x,y)* as number of mud particles on node (*x,y*), *T(x,y)* is temperature on node *(x,y)* and elementary rules in update state:

(1) The mud blow in center point (c_x,c_y) with mud volume *vol* as shown in fig. 6 that is written by:

$$s_{t+1}\left(c_x+\delta_x, c_y+\delta_y\right)=s_t(c_x, c_y)+vol.G\left(\delta_x, \delta_y\right)$$
$$T_{i+1}(c_x+\delta_x, c_y+\delta_y)=T_0 \tag{2}$$

Where:
(δ_x, δ_y) is a neighbor points
$G(\delta_x, \delta_y)$ is Gaussian based function of mud blow.
$\sum_{\delta_x}\sum_{\delta_y} G(vol, \delta_x, \delta_y)=1$
T_0 is initial temperature on center area of mud blow. We use uniform temperature on center area.

(2) Mud is situation at every lattice point. The mudflows from a higher position to lower neighborhood with probability *Pv* as the function of height different, volume and velocity as is shown in fig. 7. The number of moving mud, which is based on this rule, can be expressed by the following equation.
$m(x,y)=h_t(x,y)+s_t(x,y)$ is total of ground height and mud height in time *t*.

$$\begin{aligned} & s_t(x,y)>0, m(x+\delta_x, y+\delta_y)<m(x,y) \\ & \Rightarrow s_{t+1}\left(x+\delta_x, y+\delta_y\right)=s_t\left(x+\delta_x, y+\delta_y\right)+\varepsilon, \\ & \quad s_{t+1}(x,y)=s_t(x,y)-\varepsilon \\ & T_{t+1}(x,y)=T_t(x,y)-d(m(x,y),\varepsilon) \end{aligned} \tag{3}$$

Where: $\varepsilon=p_v.D.\left(1-\tau\left(x+\delta_x, y+\delta_y\right)\right)$

p_v = Probability to move

$D=m(x,y)-m(x+\delta_x, y+\delta_y)$

d(m,ε) is function of heat transfer.

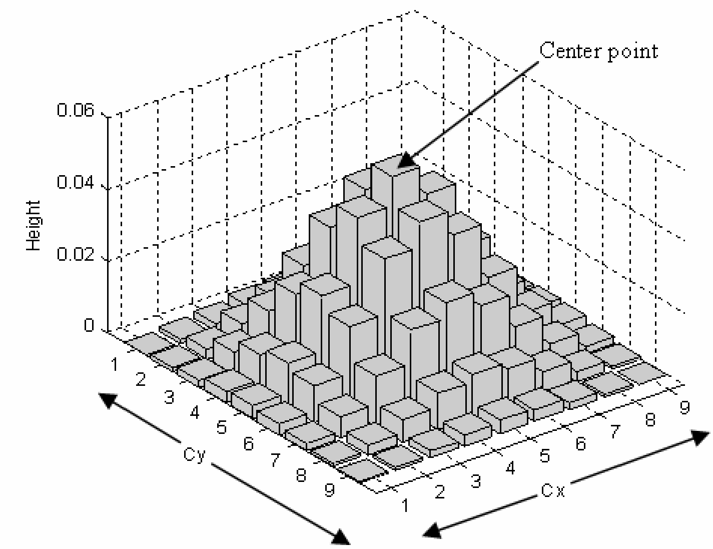


Fig. 6. Mud blow in center point (c_x,c_y) and its neighbor

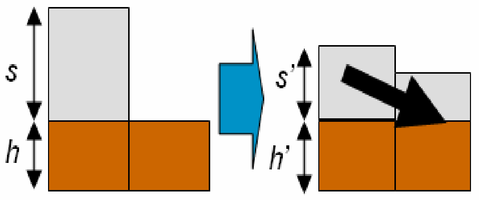


Fig. 7. Moving rules

(3) Mud changed into solid particles by p_{vis} of the probability as the function of viscosity as shown in fig. 8. The number of moving mud which is based on this rule can be represented with the following equation:

$$s_{t+1}(x,y)=s_t(x,y)-\alpha,\ h_{t+1}(x,y)=h_t(x,y)+\alpha \quad (4)$$

Where:

$\alpha = p_{vis}.(1-p_v)^2.p_T/10$

$p_T = 2-e^{-KT_t(x,y)}$

K is constants,

$T_t(x,y)$ is temperature on node (x,y) at time t.

(4) When the neighbor is dike that have higher position that mud with probability absorption P_a the mud throughout into dike and will appear in the next of dike position as shown in fig. 9.

$m(x,y)=h_t(x,y)+s_t(x,y)$ is total of ground height and mud height in time t.

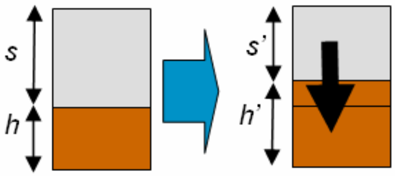


Fig. 8. Viscosity rules

$$
\begin{aligned}
& s_t(x,y) > 0, m(x+\delta_x, y+\delta_y) > m(x,y) \\
& \Rightarrow s_{t+1}(x+\delta_x+u_x, y+\delta_y+u_y) = s_t(x+\delta_x+u_x, y+\delta_y+u_x) + \varepsilon, \\
& \quad s_{t+1}(x,y) = s_t(x,y) - \varepsilon
\end{aligned}
\tag{5}
$$

Where:

$\varepsilon = p_v.p_a.D.(1-\tau(x+\delta_x, y+\delta_y))$

$(\delta_x+u_x, \delta_y+u_y)$ is node around the neighbor (δ_x, δ_y)

p_a = Probability of absorption

Based on minimization rules, the rules for hot mudflow using the material transport in moving rule and material changing in viscosity rules is defined as following rules, as shown in fig. 10:

a. *E(1)>H(0), E(2)>H(0)* → cell 1 and 3 eliminated
b. *Er*=31/3=10.3, *E(4)<Er* → cell 4 eliminated
c. *Er*=20/2=10, no cell eliminated
d. Reconfiguration □*H*=0.5*(1+ε)*(*Er-min(E)*)=3.5 → *E(0)*=7.5; *E(2)*=9.5.

Where ε is material transport based on the moving rule and viscosity rule.

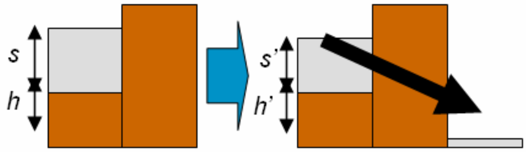


Fig. 9. Absorption rule

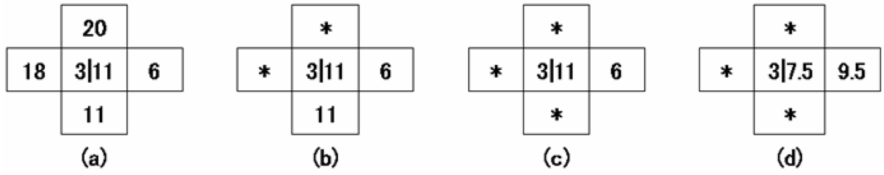


Fig. 10. Update rules for hot mud flow disaster

3.3 Data Specifications

We use SPOT-5 of HRV (Satellite Pour l'Observation de la Terre-5/High Resolution Visible) image as a base map as is shown in fig. 11(a). HRV image approximate 3.705km×4.035km area and resolution 16.46m×16.46m. We also use ASTER/DEM data for determination the landscape of intensive study area. The spatial resolution of ASTER/DEM is 30m×30m. SPOT-5/HRV image have good enough spatial resolution for relief the intensive study area. It, however, does not have well information of landscape so that ASTER/DEM data is used for creation of landscape. Re-quantization and interpolation between ASTER/DEM and some height of dike derived from SPOT-5/HRV are required. The resultant images are shown in fig. 11b. In this research we use data on February-August 2008.

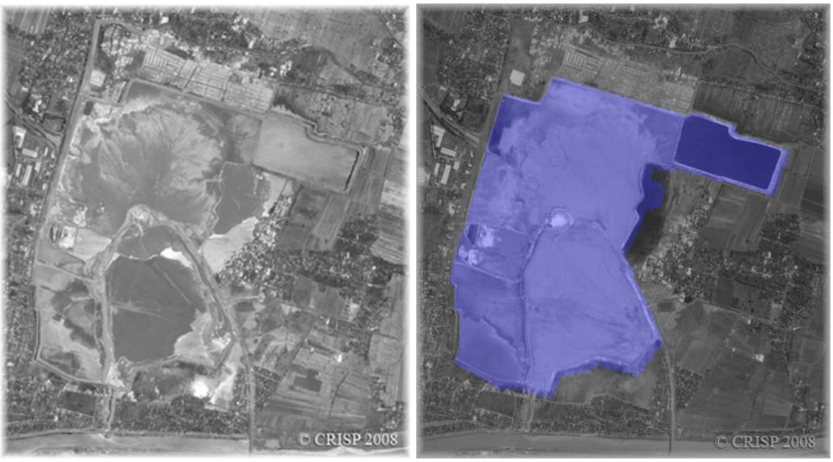
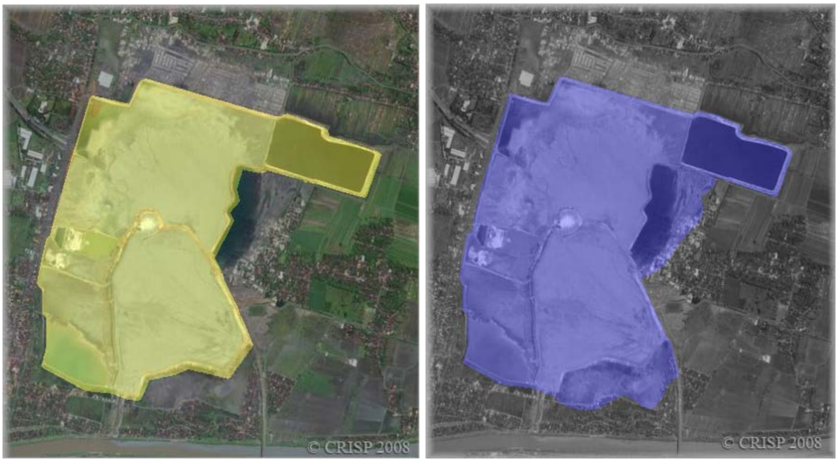


Fig. 11. (a) SPOT 5 image, (b) Landscape map image after quantization

4 Simulation Result

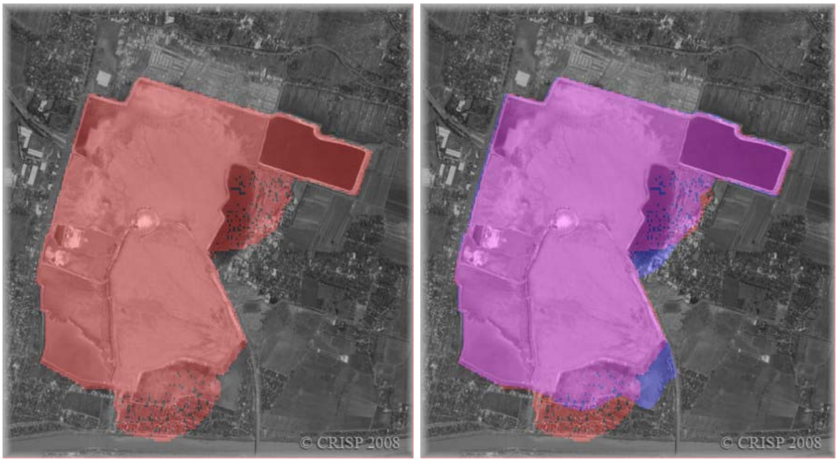
In order to find inundated area in the future, CA parameters such as spatial resolution and volume scaling have to be optimized for making the prediction much better. We use SPOT-5/HRV data in order to show the disaster area clearly. Figure 12(a) and fig. 12(b) show map of actual disaster area in February 2008 and August 2008 while fig. 13(a) shows its simulation result. The simulation begins with map of actual disaster area in February 2008.

Fig. 13(b) shows the comparison between the actual disaster area maps colored in blue area while the simulation result colored in red area, respectively. The magenta colored area shows the intersection between map of actual disaster area map and the simulation result. In this figure, although there are some different areas between map of actual disaster area and the simulation result, the direction of mudflow and inundated area are quite similar between both. Although the simulation result that is shown in the figure is derived from CA, it is possible to create a new model of mudflow with some other adding parameters such as dike and mud parameters.



(a) (b)

Fig. 12. Real Map Data: a) February 2008 b) August 2008



(a) (b)

Fig. 13. Simulation result for August 2008: a) Predicted Area b) Overlay with real map

On this simulation, we find the same inundated location on the outside of dike. The inundated location in our simulation result is on the east and south that same with the real condition of hot mudflow disaster. The intersection of the new inundated area between real map and simulation result is 36.44% as shown fig. 14. Although the intersection of inundated area are not high, but our simulation results have same direction with the real condition, it means our method can be show the direction of mudflow, and it needs higher resolution map and some additional formula to make the result better such as mixing-ratio pattern of solid and water, thermal changing, water absorption and land-use data.

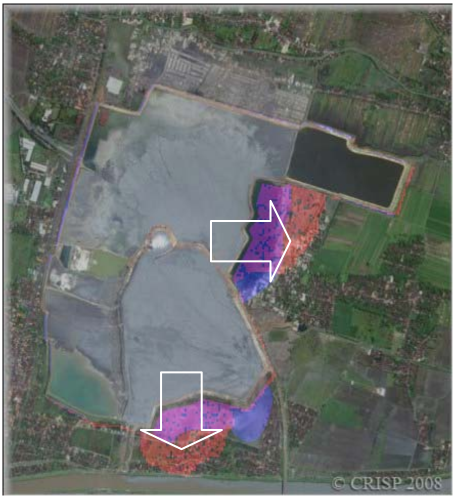


Fig. 14. Comparison between simulation result and actual disaster map

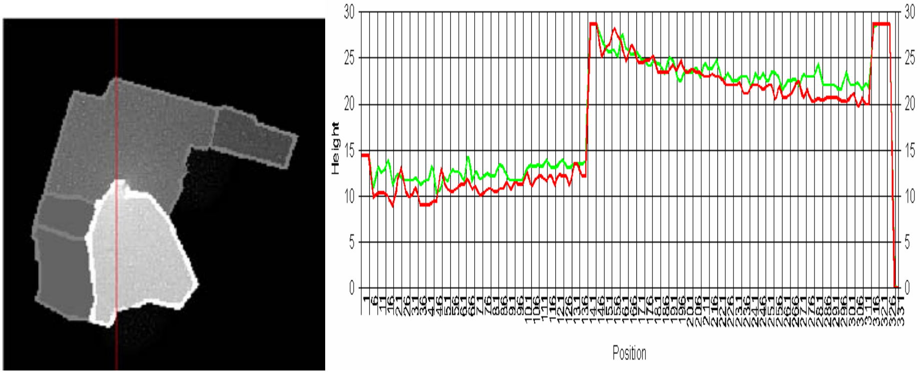


Fig. 15. Cutting line to show mud height changing and mud elevation at every point along with the line from the center point on the right image

Another simulation result of elevation changes at hot mudflow erupted areas. For this result, we show the mud height changing on the red line cutting (vertical) as shown in fig. 15 (left). Those areas are the main area of mud flow, and if we can show increasing of the mud height, we can reduce the impact of mudflow.

This result shows that mud elevation changes depending upon the initial and the final conditions. Fig. 15 (right) shows the one-dimensional profile of the mud elevation changes. The red line shows the mud elevation at the initial state that we take the landscape data of February 2008. The green line shows the mud elevation at the final state, three months later from the initial state. The average elevation difference between the initial and the final state is 1.01meter. It implies that the mud elevation changes about 0.3meter per-month.

5 Discussion and Conclusion

Cellular Automata approach is a model-based approach that depends on some parameters such as resolution, neighborhood, and rules. This model is accurate when the resolution is appropriate for representation of particles or cells. Meanwhile the proposed model makes a relaxant on the required resolution. Even for the minimum resolution of 100×100pixels, the proposed method makes an enough simulation result (the maximum resolution is 2000×2000 pixels). The minimum resolution is corresponding to 37m×40m a pixel that is also corresponding to the lower resolution of ASTER/DEM data with 30m×30m so that the proposed method is justified and evaluated with ASTER/DEM data. The maximum resolution (800×800 pixels) of simulation result is shown here. This resolution corresponds to 4.625m×5m a pixel. It is concluded that the proposed method is valid for detection and prediction of hot mudflow spreading direction and volume as well as appropriate inundated areas that are situated surrounding areas.

References

1. Harsaputra, I.: Govt. weighs options for battling the sludge, The Jakarta Post, Comments and Prospects, May 29 (2007), `http://mudflow-sidoarjo.110mb.com` (retrieved 11.06.07)
2. Sjahroezah, A.: Environmental Impact of the hot mud flow in Sidoarjo, East Java. The SPE Luncheon Talk, April 19 (2007)
3. Argentini, G.: A first approach for a possible cellular automaton model of fluids dynamic. New Technologies & Models, Information & Communication Technology Department Riello Group, Legnago (Verona), Italy (February 2003)
4. Vicari, A., Alexis, H., Negro, C.D., Coltelli, M., Marsella, M., Proietti, C.: Modeling of the 2001 Lava Flow at Etna Volcano by a Cellular Automata Approach. Environmental Modelling & Software 22, 1465–1471 (2007)
5. Emanuel, G.: Analytical fluid dynamics, 2nd edn., pp. 6–7. CRC Press, Boca Raton (2001)
6. Margolus, N., Toffoli, T., Vichniac, G.: Cellular Automata Supercomputers for Fluid-Dynamics Modeling. The American Physical Review Letters 56(16) (April 21, 1986)
7. Chopard, B., Luthi, P., Masselot, A.: Cellular Automata and Lattice Boltzmann Techniques: An Approach to Model and Simulate Complex Systems. In: Proceeding of Advances in Physics Conference (1998)
8. Ghazali, J.N., Kamsin, A.: A Real Time Simulation of Flood Hazard. In: Fifth International Conference on Computer Graphics, Imaging and Visualization, pp. 393–397. IEEE, Los Alamitos (2008)
9. Yu, T., Lee, S.: Evolving Cellular Automata To Model Fluid Flow In Porous Media. In: Proceedings NASA/DoD Conference on Evolvable Hardware, pp. 210–217 (2002)
10. Avolio, M.V., Di Gregorio, S., Mantovaniz, F., Pasuto, A., Rongo, R., Silvano, S., Spataro, W.: Simulation of the 1992 Tessina landslide by a cellular automata model and future hazard scenarios. JAG l 2(1), 41–50 (2000)
11. D'Ambrosio, D., Di Gregorio, S., Gabriele, S., Gaudio, R.: A Cellular Automata Model for Soil Erosion by Water. Physis and Chemistry of the Earth, EGS, B 26(1), 33–39 (2001)
12. Rinaldi, P.R., Dalponte, D.D., Veénere, M.J.: Alejandro Clausse, Cellular automata algorithm for simulation of surface flows in large plains. Simulation Modelling Practice and Theory 15, 315–327 (2007)

Stochastic Optimization Approaches to Image Reconstruction in Electrical Impedance Tomography

Chang-Jin Boo[1], Ho-Chan Kim[1], Min-Jae Kang[2], and Kwang Y. Lee[3]

[1] Department of Electrical Engineering, Jeju National University, Korea
{boo1004,hckim}@jejunu.ac.kr
[2] Department of Electronic Engineering, Jeju National University, Korea
minjk@jejunu.ac.kr
[3] Department of Electrical and Computer Engineering, Baylor University, USA
Kwang_Y_Lee@baylor.edu

Abstract. In electrical impedance tomography (EIT), various image reconstruction algorithms have been used in order to compute the internal resistivity distribution of the unknown object with its electric potential data at the boundary. Mathematically the EIT image reconstruction algorithm is a nonlinear ill-posed inverse problem. This paper presents two stochastic optimization techniques such as particle swarm optimization (PSO).and simultaneous perturbation stochastic approximation (SPSA) algorithms for solving the static EIT inverse problem. We summarize the simulation results for the three algorithm forms: modified Newton-Raphson, particle swarm optimization, and simultaneous perturbation stochastic approximation.

1 Introduction

Electrical impedance tomography (EIT) plays an important role as a new monitoring tool for engineering applications such as biomedical imaging and process tomography, due to its relatively cheap electronic hardware requirements and nonintrusive measurement property [1]. In EIT, different current patterns are injected to the unknown object through electrodes and the corresponding voltages are measured on its boundary surface. The physical relationship between inner resistivity (or conductivity) and boundary surface voltage is governed by the nonlinear Laplace equation with appropriate boundary conditions, so that it is impossible to obtain the closed-form solution for the resistivity distribution. Hence, the internal resistivity distribution of the unknown object is computed using the boundary voltage data based on various reconstruction algorithms.

Yorkey et al. [2] developed a modified Newton-Raphson (mNR) algorithm for a static EIT image reconstruction and compared it with other existing algorithms such as backprojection, perturbation and double constraints methods. They concluded that the mNR reveals relatively good performance in terms of convergence rate and residual error compared to those of the other methods. However, in real situations, the mNR method has often failed to obtain satisfactory images from physical data due to large modeling error, poor signal to noise ratios (SNRs) and ill-conditioned (ill-posed) characteristics. That is, the ratio between the maximum and minimum eigenvalues of

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 99–109, 2010.

the information matrix (or Hessian matrix) is very large. In particular, the ill-conditioning of the information matrix results in an inaccurate matrix inverse such that the resistivity update process is very sensitive to the modeling and measurement errors.

The major difficulties in impedance imaging are in the nonlinearity of the problem itself and the poor sensitivity of the boundary voltages to the resistivity of the flow domain deep inside. Several researchers suggested various element or mesh grouping methods where they force all meshes belonging to certain groups to have the same resistivity values [3,4].

In this paper, we will discuss the image reconstruction based on stochastic optimization approaches in EIT. We have broken the procedure for obtaining the internal resistivity distribution into two steps. In the first step, each mesh is classified into three mesh groups: target, background, and temporary groups. In the second step, the values of these resistivities are determined using particle swarm optimization (PSO) [5-12] and simultaneous perturbation stochastic approximation (SPSA) [13-16] algorithms. This two-step approach allows us to better constrain the inverse problem and subsequently achieve a higher spatial resolution.

2 Image Reconstruction Using Stochastic Optimization Approaches in EIT

The numerical algorithm used to convert the electrical measurements at the boundary to a resistivity distribution is described here. The algorithm consists of iteratively solving the forward problem and updating the resistivity distribution as dictated by the formulation of the inverse problem. The forward problem of EIT calculates boundary potentials with the given electrical resistivity distribution, and the inverse problem of EIT takes potential measurements at the boundary to update the resistivity distribution.

2.1 Forward Problem

When electrical currents $I_l(l=1,\cdots,L)$ are injected into the object $\Omega\in R^2$ through electrodes $e_l(l=1,\cdots,L)$ attached on the boundary $\partial\Omega$ and the resistivity distribution $\rho(x,y)$ is known over Ω, the corresponding induced electrical potential $u(x,y)$ can be determined uniquely from the nonlinear Laplace equation which can be derived from the Maxwell equation, Ohm's law, and the Neumann type boundary condition. The complete electrode model takes into account both the shunting effect of the electrode and the contact impedances between the electrodes and the object. The equations of complete electrode model [17] are

$$\nabla\cdot(\rho^{-1}\nabla u)=0 \text{ in } \Omega \tag{1}$$

$$\begin{aligned}
&\int_{e_l} \rho^{-1} \frac{\partial u}{\partial n} ds = I_l, \ l = 1, \cdots, L \\
&u + z_l \rho^{-1} \frac{\partial u}{\partial n} = U_l \ \text{ on } e_l, \ l = 1, \cdots, L \\
&\rho^{-1} \frac{\partial u}{\partial n} = 0 \ \text{ on } \partial\Omega \backslash \bigcup_{l=1}^{L} e_l
\end{aligned} \tag{2}$$

where z_l is effective contact impedance between the l th electrode and the object, U_l is the measured potential at the l-th electrode and n is outward unit normal. In addition, we have the following two conditions for the injected currents and measured voltages by taking into account the conservation of electrical charge and appropriate selection of ground electrode, respectively.

$$\sum_{l=1}^{L} I_l = 0, \ \sum_{l=1}^{L} U_l = 0 \tag{3}$$

The computation of the potential $u(x, y)$ for the given resistivity distribution $\rho(x, y)$ and boundary condition I_l is called the *forward problem.* The numerical solution for the forward problem can be obtained using the finite element method (FEM). In the FEM, The potential at each node is calculated by discretizing (1) into $Yu = c$, where u is the vector of boundary potential, c the vector of injected current patterns and the matrix Y is a functions of the unknown resistivities.

2.2 Inverse Computation Using Stochastic Optimization Approaches

The inverse problem, also known as the image reconstruction, consists in reconstructing the resistivity distribution $\rho(x, y)$ from potential differences measured on the boundary of the object. Ideally, knowing the potential on the whole boundary makes the correspondence between the resistivity distribution and the potential biunique. The relatively simple situation depicted so far does not hold exactly in the real world. The methods used for solving the EIT problem search for an approximate solution, i.e., for a resistivity distribution minimizing some sort of residual involving the measured and calculated potential values. From a mathematical point of view, the EIT inverse problem consists of finding the coordinates of a point in a M -dimensional hyperspace, where M is the number of discrete elements whose union constitutes the tomographic section under consideration. In the past, several EIT image reconstruction algorithms for the current injection method have been developed by various authors. A review of these methods is given in [18]. To reconstruct the resistivity distribution inside the object, we have to solve the nonlinear ill-posed inverse problem. Regularization techniques are needed to weaken the ill-posedness and to obtain stable solutions. Generalized Tikhonov regularized version of the EIT inverse problem can be written in the form [17]

$$E(\rho) = \min_{\rho}\{ \| U - V(\rho) \|^2 + \lambda \| R\rho \|^2 \} \tag{4}$$

where $\rho \in R^N$ is the resistivity distribution. $V(\rho)$ is the vector of voltages obtained from the model with known ρ , U are the measured voltages and R and λ are the regularization matrix and the regularization parameter, respectively. There are many approaches in the literature [19] to determine R and α, but the usual choice is to fix $R = I_M$ with the identity matrix and to adjust λ empirically.

Modified Newton-Raphson Algorithm [20]. Minimizing the objective function $E(\rho)$ gives an equation for the update of the resistivity vector

$$\begin{aligned} \rho_{k+1} &= \rho_k + \Delta\rho_{k+1} \\ \Delta\rho_{k+1} &= (H_k + \lambda I)^{-1}\{J_k{}^T(U - V(\rho_k)) - \lambda\rho_k\} \end{aligned} \tag{5}$$

where the partial derivative of E with respect to ρ has been approximately by a Taylor series expansion around ρ_k . The Jacobian J_k is a matrix composed of the derivative of the vector of predicted potentials with respect to the unknown resistivities. The Jacobian is derived from the finite element formulation given by $J_k = \left.\frac{\partial E}{\partial \rho}\right|_{\rho_k}$. The Hessian H_k is the second derivative of the predicted potentials with respect to the resistivity and is approximated as the square of the Jacobian for computational efficiency. Since the objective function $E(\rho)$ is multimodal (i.e., it presents several local minima), the inversion procedure does not always converge to the true solution. The reconstruction algorithms are likely to be trapped in a local minimum and sometimes the best solution of a static EIT problem is rather unsatisfactory.

This paper attempts to apply stochastic optimization approaches such as PSO and SPSA to EIT image reconstruction. Two characteristics of PSO and SPSA algorithms appear to be of value in EIT reconstruction; no evaluation of function derivatives is needed and no assumption on function continuity needs to be made. The preceding considerations suggest the viability of employing stochastic optimization approaches for the solution of the EIT problem, according to the procedure described in the following section.

Furthermore, in some applications like visualization of two-component systems, we may assume that there are only two different representative resistivity values; one resistivity value for the background and the other for the target. Here, the target need not be a single segment. It may be composed of multiple segments of the same resistivity value.

In this paper, we will discuss the image reconstruction in EIT using two-step approach. We have broken the procedure for obtaining the internal resisitivity distribution into two steps. In the first step, we adopted a mNR method as a basic image reconstruction algorithm. After a few initial mNR iterations performed without any grouping, we classify each mesh into one of three mesh groups: BackGroup and TargetGroup are the mesh groups with the resistivity values of the background and target, respectively. TempGroup is the group of meshes neither in BackGroup nor in TargetGroup. All meshes in BackGroup and in TargetGroup are forced to have the same but unknown resistivity values, ρ_{back} and ρ_{tar} , respectively. However, all meshes in TempGroup can have different resistivity values, $\rho_{temp,i}$, $i = 1, \cdots, n-2$.

PSO Algorithm [6]. Kennedy and Eberhart [5] developed a PSO algorithm based on the behavior of individuals (i.e., particles or agents) of a swarm. Its roots are in zoologist's modeling of the movement of individuals (e.g., fishes, birds, or insects) within a group. It has been noticed that members within a group seem to share information among them, a fact that leads to increased efficiency of the group. The PSO algorithm searches in parallel using a group of individuals similar to other AI-based heuristic optimization techniques [6]. An individual in a swarm approaches to the optimum or a quasi-optimum through its present velocity, previous experience, and the experience of its neighbors. The main advantages of the PSO algorithm are summarized as: simple concept, easy implementation, robustness to control parameters, and computational efficiency when compared with mathematical algorithm and other heuristic optimization techniques. Due to these advantages, several variations of the PSO have been developed and applied in power system optimization problems [7-12].

In a physical n -dimensional search space, the position and velocity of an individual i are represented as the vectors $X_i = (x_{i1}, \cdots, x_{in})$, and $V_i = (v_{i1}, \cdots, v_{in})$, respectively, in the PSO algorithm. Let $\mathrm{Pbest}_i = (x_{i1}^{\mathrm{Pbest}}, \cdots, x_{in}^{\mathrm{Pbest}})$, and $\mathrm{Gbest}_i = (x_{i1}^{\mathrm{Gbest}}, \cdots, x_{in}^{\mathrm{Gbest}})$, respectively, be the best position of an individual i and its neighbors' best position so far. Using the information, the updated velocity of individual i is modified under the following equation in the PSO algorithm:

$$V_i^{k+1} = \omega V_i^k + c_1 \mathrm{rand}_1 \times (\mathrm{Pbest}_i^k - X_i^k) + c_2 \mathrm{rand}_2 \times (\mathrm{Gbest}_i^k - X_i^k) \qquad (6)$$

where

V_i^k	velocity of individual i at iteration k ;
ω	weight parameter;
c_1, c_2	weight factors;
$\mathrm{rand}_1, \mathrm{rand}_2$	random numbers between 0 and 1;
X_i^k	position of individual i at iteration k ;
Pbest_i^k	best position of individual i until iteration k ;;
Gbest_i^k	best position of the group until iteration k .

Each individual moves from the current position to the next one by the modified velocity in (6) using the following equation:

$$X_i^{k+1} = X_i^k + V_i^{k+1} \qquad (7)$$

In this velocity updating process, the values of parameters such as ω, c_1 , and c_2 should be determined in advance. In general, the weight ω is set according to the following equation [7]:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{Iter_{\max}} \times Iter \tag{8}$$

where

$\omega_{\max}$, $\omega_{\min}$ initial, final weights,

$Iter_{\max}$ maximum iteration number,

$Iter$ current iteration number.

The step-by-step process of the proposed PSO algorithm can be summarized as follows:

Step 1) Initialization of a group at random while satisfying constraints.
Step 2) Velocity and position updates while satisfying constraints.
Step 3) Update of *Pbest* and *Gbest*.
Step 4) Activation of space reduction strategy.
Step 5) Go to Step 2 until satisfying stopping criteria.

SPSA Algorithm [13]. Spall developed the concept of simultaneous perturbation stochastic approximation (SPSA) as an optimization tool. The SPSA algorithm works by iterating from an initial guess of the optimal solution, where the iteration process depends on the highly efficient "simultaneous perturbation" approximation to the gradient of the objective function. The SPSA algorithm has been applied in power systems [14,15] as well as in the EIT problem [16]. The SPSA reconstruction algorithm for EIT can be formulated as follows. The goal is to minimize a loss function $L(\rho)$, where the loss function is a scalar-valued "performance measure" and ρ is a continuous-valued p-dimensional vector of parameters to be adjusted. The SPSA algorithm works by iterating from an initial guess of the optimal, where the iteration process depends on the above-mentioned "simultaneous perturbation" approximation to the gradient $g(\rho) \equiv \partial L(\rho) / \partial \rho$. Assume that measurements of the loss function are available at any value of ρ :

$$E(\rho) = L(\rho) + noise, \quad L(\rho) = \min_{\rho}\{\left\| \frac{U - V(\rho)}{V(\rho)} \right\|\} \tag{9}$$

The basic unconstrained SPSA algorithm is in the general recursive stochastic approximation (SA) form

$$\rho_{k+1} = \rho_k - a_k g_k(\rho_k) \tag{10}$$

where $g_k(\rho_k)$ is the simultaneous perturbation estimate of the gradient $g(\rho)$ at the iterate ρ_k based on the measurements of the loss function and a_k is a nonnegative scalar gain coefficient.

The essential part of (10) is the gradient approximation $g_k(\rho_k)$. This gradient approximation is formed by perturbing the components of ρ_k one at a time and collecting a loss measurement $E(\cdot)$ at each of the perturbations (in practice, the loss measurements are sometimes noise-free, $E(\cdot)=L(\cdot)$). This requires $2p$ loss measurements for a two-sided finite difference approximation. All elements of ρ_k are randomly perturbed together to obtain two loss measurements $E(\cdot)$. For the two-sided simultaneous perturbation gradient approximation, this leads to

$$g_k(\rho_k) = \frac{E(\rho_k + c_k\Delta_k) - E(\rho_k - c_k\Delta_k)}{2c_k}\begin{bmatrix}\Delta_{k1}^{-1}\\ \Delta_{k2}^{-1}\\ \vdots \\ \Delta_{kp}^{-1}\end{bmatrix} \tag{11}$$

where the mean-zero p-dimensional random perturbation vector, $\Delta_k = [\Delta_{k1}, \Delta_{k2}, \cdots, \Delta_{kp}]^T$, has a user-specified distribution and c_k is a positive scalar. Because the numerator is the same in all p components of $g_k(\rho_k)$, the number of loss measurements needed to estimate the gradient in SPSA is two, regardless of the dimension p.

The step-by-step summary below shows how SPSA iteratively produces a sequence of estimates.

Step 1 Initialization and coefficient selection**:** Set counter index k=0. Pick initial guess ρ_0 in (5) and the nonnegative coefficients a, c, A, α , and γ in the SPSA gain sequences $a_k = a/(A+k+1)^{\alpha}$ and $c_k = c/(k+1)^{\gamma}$. Practically effective values for α and γ are 0.602 and 0.101, respectively.

Step 2 Generation of simultaneous perturbation vector**:** Generate by Monte Carlo a p-dimensional random perturbation vector Δ_k, where each of the p components of Δ_k are independently generated from a zero-mean probability distribution satisfying the conditions in Spall [13]. A simple choice for each component of Δ_k is to use a Bernoulli ± 1 distribution with probability of 0.5 for each ± 1 outcome.

Step 3 Loss function evaluations**:** Obtain two measurements of the loss function based on the simultaneous perturbation around the current ρ_k : $E(\rho_k + c_k\Delta_k)$ and $E(\rho_k - c_k\Delta_k)$ in (9) with the c_k and Δ_k from Step 1 and 2.

Step 4 Gradient approximations**:** Generate the simultaneous perturbation approximation to the unknown gradient $g_k(\rho_k)$ according to (11). It is sometimes useful to average several gradient approximations at ρ_k, each formed from an independent generation of Δ_k.

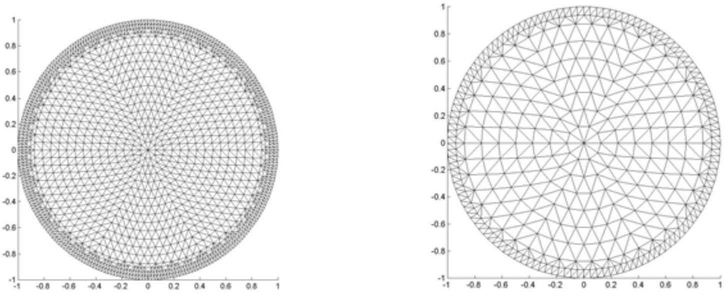
Step 5 Updating parameter ρ **:** Use the standard stochastic approximation form in (10) to update ρ_k to a new value ρ_{k+1}.

Step 6 Iteration or Termination**:** Return to Step 2 with $k+1$ replacing k. Terminate the algorithm if there is little change in several successive iteration or the maximum allowable number of iterations has been reached.

The choice of a_k and c_k is critical to the performance of SPSA. With α and γ as specified in Step 1, one typically finds that in a high-noise setting it is necessary to pick a smaller a and larger c than in a low-noise setting. Although the asymptotically optimal values of α and γ are 1.0 and $1/6$, respectively, it appears that choosing $\alpha < 1.0$ usually yields better finite-sample performance through maintaining a larger step size; hence the recommendation in Step 1 to use values (α and γ) that are effectively the lowest allowable satisfying the theoretical conditions mentioned [13].

3 Computer Simulation

The proposed algorithms have been tested by comparing its results for numerical simulations with those obtained by mNR method. For the current injection the trigonometric current patterns were used. For the forward calculations, the domain Ω was a unit disc and the mesh of 3104 triangular elements (M=3104) with 1681 nodes (N=1681) and 32 channels (L=32) was used as shown in Fig. 1(a). A different mesh system with 776 elements (M=776) and 453 nodes (N=453) was adopted for the inverse calculations as shown in Fig. 1(b). In this paper, under the assumption that the resistivity varies only in the radial direction within a cylindrical coordinate system [21], the results of the three inverse problem methods can be easily compared. The resistivity profile given to the finite element inverse solver varies from the center to the boundary of object and is divided into 9 radial elements ($\rho_1, \cdots, \rho_9$) as shown in Fig. 1(b).



(a) (b)

Fig. 1. Finite element mesh used in the calculation. (The resistivities of the elements within an annular ring are identical.) (a) mesh for forward solver, (b) mesh for inverse solver.

Synthetic boundary potentials were computed for idealized resistivity distributions using the finite element method described earlier. The boundary potentials were then used for inversion and the results were compared to the original resistivity profiles. The resistivity profile appearing in Fig. 2 contains two large discontinuities in the original resistivity distribution. The present example is a severe test in EIT problems because there are large step changes at r/R=0.56 and 0.81 preventing electric currents from going into the center region.

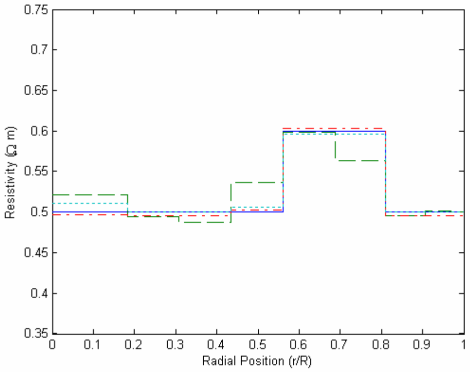


Fig. 2. True resistivities(solid line) and coumputed resistivities using mNR(dashed line), PSO(dashdot line), and SPSA(dotted line)

Table 1. True resistivities and computed resistivities using mNR, PSO and SPSA

	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	ρ_9
Real	0.5	0.5	0.5	0.5	0.6	0.6	0.5	0.5	0.5
mNR	.521	.495	.488	.537	.598	.564	.496	.502	.500
PSO	.497	. 496	. 496	.503	.603	. 603	. 496	. 496	. 496
SPSA	.511	.500	.500	.506	.596	.596	.500	.500	.500

We started the mNR iteration without any mesh grouping with a homogeneous initial guess. In Table 1, we see that the mNR algorithm may roughly estimate the given true resistivities, where the resistivities are assumed uniform in each section. Since the mNR have a large error at the boundary of target and background in Fig. 2, we can not obtain reconstructed images of high spatial resolution. This kind of poor convergence is a very typical problem in the NR-type algorithms.

However, we can significantly improve the mNR's poor convergence by adopting the proposed stochastic optimization algorithms such as PSO and SPSA via a two-step approach as follows. In the first step, we adopted a mNR method as a basic image reconstruction algorithm. After a few initial mNR iterations performed without any grouping, we rearrange the resisitivity values of meshes by sorting them in ascending order. Then the boundary location between regions can be roughly decided and mesh be determined to the target, background, or undetermined temporary group. In this paper, from the Table 1, 2 meshes (ρ_5, ρ_6) and 5 meshes ($\rho_2, \rho_3, \rho_7, \rho_8, \rho_9$) among 9 may be grouped to TargetGroup (ρ_{tar}) and BackGroup (ρ_{back}), respectively. The remainders of meshes (ρ_1, ρ_4) are grouped to TempGroup. Hence, the number of unknowns is reduced to 4.

In the second step, after mesh grouping, we will determine the values of these resistivities using PSO and SPSA algorithms. PSO and SPSA algorithms solve the EIT problem, searching for the resistivities ($\rho_1, \rho_4, \rho_{tar}$ and ρ_{back}) minimizing the reconstruction error. The initial values of unknown ρ_{tar} and ρ_{back} are the average resistivity values of meshes in BackGroup and TargetGroup, respectively. From Fig. 2 and Table 1, the inverted profile using PSO and SPSA matches the original profile very well near the wall at r/R =1.0 as well as the center at r/R =0.0.

4 Conclusion

In this paper, electrical impedance tomography (EIT) image reconstruction methods based on stochastic optimization algorithms were presented to improve the spatial resolution. A technique based on two stochastic optimization algorithms, PSO and SPSA, with the knowledge of mNR, was developed for the solution of the EIT inverse problem. Although stochastic optimization algorithms such as PSO and SPSA are expensive in terms of computing time and resources, which is a weakness of the method that renders it presently unsuitable for real-time tomographic applications, the exploitation of a priori knowledge will produce very good reconstructions. Further extensions include an EIT image reconstruction to multi-resistivity value problems.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0077743)

References

1. Webster, J.G.: Electrical Impedance Tomography. Adam Hilger (1990)
2. Yorkey, T.J., Webster, J.G., Tompkins, W.J.: Comparing Reconstruction Algorithms for Electrical Impedance Tomography. IEEE Trans. on Biomedical Engineering. 34, 843–852 (1987)
3. Glidewell, M., Ng, K.T.: Anatomically Constrained Electrical Impedance Tomography for Anisotropic Bodies Via Two-step Approach. IEEE Trans. on Medical Imaging 14, 498–503 (1995)
4. Paulsen, K.D., Meaney, P.M., Moskowitz, M.J., Sullivan, J.M.: A Dual Mesh Scheme for Finite Element Based Reconstruction Algorithm. IEEE Trans. on Medical Imaging 14, 504–514 (1995)
5. Kennedy, J., Eberhart, R.: Swarm Intelligence. Morgan Kaufmann Publishers, Inc, San Francisco (2001)
6. Lee, K.Y., El-Sharkawi, M.A. (Editors): Modern Heuristic Optimization Techniques with Applications to Power Systems. Wiley, New York (2008)
7. Park, J.-B., Lee, K.-S., Shim, J.-R., Lee, K.Y.: A Particle Swarm Optimization for Economic Dispatch with Nonsmooth Cost Functions. IEEE Trans. on Power Systems 20, 34–42 (2005)
8. Vlachogiannis, J.G., Lee, K.Y.: Determining Generator Contributions to Transmission System Using Parallel Vector Evaluated Particle Swarm Optimization. IEEE Transactions on Power Systems 20, 1765–1774 (2005)

9. Heo, J.S., Lee, K.Y., Garduno-Ramirez, R.: Multiobjective Control of Power Plants using Particle Swarm Optimization Techniques. IEEE Transactions on Energy Conversion 21, 552–561 (2006)
10. Vlachogiannis, J.G., Lee, K.Y.: A Comparative Study on Particle Swarm Optimization for Optimal Steady-State Performance of Power Systems. IEEE Transactions on Power Systems 21, 1718–1728 (2006)
11. Park, J.-B., Jeong, Y.-W., Shin, J.-R., Lee, K.Y., Kim, J.-H.: A Hybrid Particle Swarm Optimization Employing Crossover Operation for Economic Dispatch Problems with Valve-point Effects. Engineering Intelligent Systems for Electrical Engineering and Communications 15, 69–74 (2007)
12. Vlachogiannis, J.G., Lee, K.Y.: Economic Dispatch – A Comparative Study on Heuristic Optimization Techniques with an Improved Coordinated Aggregation-Based PSO. IEEE Transactions on Power Systems 24, 991–1001 (2005)
13. Spall, J.C.: Genetic Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control. Wiley, Chichester (1989)
14. Ko, H.-S., Lee, K.Y., Kim, H.-C.: Electricity Price Prediction Model Based on Simultaneous Perturbation Stochastic Approximation. Journal of Electrical Engineering & Technology 3, 14–19 (2008)
15. Ko, H.-S., Lee, K.Y., Kim, H.-C.: A Simultaneous Perturbation Stochastic Approximation (SPSA)-Based Model Approximation and its Application for Power System Stabilizers. International Journal of Control, Automation, and Systems 6, 506–514 (2008)
16. Kim, H.C., Boo, C.J., Lee, Y.J.: A SPSA Approach to Image Reconstruction in Electrical Impedance Tomography. J. of the Korean Institute of Illuminating and Electrical Installation Engineers. 3, 23–28 (2004)
17. Vauhkonen, M.: Electrical Impedance Tomography and Priori Information. Kuopio Univerisity Publications Co. Natural and Environmental Sciences 62 (1997)
18. Murai, T., Kagawa, Y.: Electrical Impedance Computed Tomography Based on a Finite Element Model. IEEE Trans. on Biomedical Engineering 32, 177–184 (1985)
19. Cohen-Bacrie, C., Goussard, Y., Guardo, R.: Regularized Reconstruction in Electrical Impedance Tomography Using a Variance Uniformization Constraint. IEEE Trans. on Medical Imaging 16, 170–179 (1997)
20. Kim, H.C., Boo, C.J.: Intelligent Optimization Algorithm Approach to Image Reconstruction in Electrical Impedance Tomography. In: Jiao, L., Wang, L., Gao, X.-b., Liu, J., Wu, F. (eds.) ICNC 2006. LNCS, vol. 4221, pp. 856–859. Springer, Heidelberg (2006)
21. Kim, M.C., Kim, S., Kim, K.Y., Lee, J.H., Lee, Y.J.: Reconstruction of Particle Concentration Distribution in Annular Couette Flow Using Electrical Impedance Tomography. J. Ind. Eng. Chem. 7, 341–347 (2001)

Estimating Soil Parameters Using the Kernel Function

Min-Jae Kang[1], Chang-Jin Boo[2], Ho-Chan Kim[2], and Jacek M. Zurada[3]

[1] Department of Electronic Engineering, Jeju National University, Korea
minjk@jejunu.ac.kr
[2] Department of Electrical Engineering, Jeju National University, Korea
{boo1004,hckim}@jejunu.ac.kr
[3] Department of Electrical Engineering, University of Louisville, USA
j.zurada@ieee.org

Abstract. In this paper, a fast algorithm for estimating soil parameters has been presented according to which, after obtaining the kernel function, one can compute the soil parameters of the multilayer earth structure by analyzing the kernel function. The estimated soil parameters using the proposed method are in good agreement with the given earth structure.

1 Introduction

It is important to know the earth structure in the given area when the grounding system is designed. Because badly designed grounding system can not ensure the safety of equipment and personnel [1]. Usually the earth structure is complex since the resistivity of the soil varies with the depth and other conditions. For simplifying the problem, in a host of engineering application, multilayer soils are modeled by N horizontal layers with distinct resistivity and depths [2]. A Wenner 4-point test method is well known to measure the soil resistivity for this simplified earth model. The inversion of soil parameter and structure is an unconstrained nonlinear minimization problem [3],[4]. There exist two difficulties in inverting the soil parameters using optimization methods. On one hand, it is hard to obtain the derivatives of the optimized expression. On the other hand, the computing time is hugely consumed. In this paper, a fast algorithm is presented to invert the parameters of horizontal multilayer soil. The soil parameters are estimated analyzing the kernel function. The proposed method has been contributed to: First, it simplifies the problem by not requiring the derivatives of the optimized expression; second it can save computational time.

2 Wenner 4-Point Test and Apparent Resistivity

A Wenner 4 –point method is well known to measure the earth resistivity as shown in Fig. 1, where h_i $(i = 1, 2, \cdots, N-1)$ and ρ_i $(i = 1, 2, \cdots, N)$ are the thickness and the resistivity of the i th layer respectively for an N – layer soil structure.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 110–118, 2010.

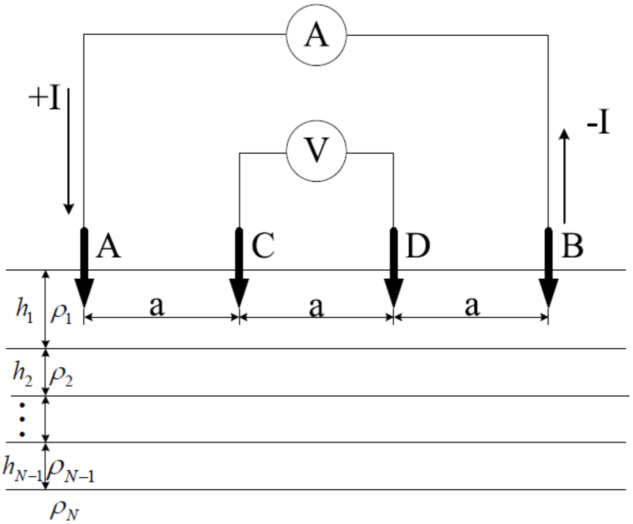


Fig. 1. Wenner configuration for measuring apparent soil resistivity of N-layer earth structure

A current I is injected into the soil by applying the power between electrodes A and B and the potential difference V between electrodes C and D is measured. Then, the measured apparent resistivity is defined [5] as

$$\rho_a^m = 2\pi a \frac{V}{I} \tag{1}$$

where a is the span between any two neighboring electrodes. By changing the test electrode span a, a set of apparent resistivity curves varying with electrode span can be obtained. And using the solution of the potential produced by point current sources of half-spherical electrodes, the theoretical expression of the apparent resistivity is as follows [1], [4]

$$\rho_a^c = \rho_1 \left\{ 1 + 2a \int_0^\infty f(\lambda)[J_0(\lambda a) - J_0(2\lambda a)] d\lambda \right\} \tag{2}$$

where $J_0(\lambda\gamma)$ is the zero order Bessel's function of the first kind and the kernel function $f(\lambda)$ is as follows [1]

$$f(\lambda) = \alpha_1(\lambda) - 1\,. \tag{3}$$

$$\begin{aligned}
\alpha_1(\lambda) &= 1 + \frac{2K_1 e^{-2\lambda h_1}}{1 - K_1 e^{-2\lambda h_1}} & K_1(\lambda) &= \frac{\rho_2\alpha_2 - \rho_1}{\rho_2\alpha_2 + \rho_1} \\
\alpha_2(\lambda) &= 1 + \frac{2K_2 e^{-2\lambda h_2}}{1 - K_2 e^{-2\lambda h_2}} & K_2(\lambda) &= \frac{\rho_3\alpha_3 - \rho_2}{\rho_3\alpha_3 + \rho_2} \\
&\vdots & &\vdots \\
\alpha_{N-1}(\lambda) &= 1 + \frac{2K_{N-1} e^{-2\lambda h_{N-1}}}{1 - K_{N-1} e^{-2\lambda h_{N-1}}} & K_{N-1}(\lambda) &= \frac{\rho_N - \rho_{N-1}}{\rho_N + \rho_{N-1}}.
\end{aligned} \tag{4}$$

The apparent resistivity for N horizontal earth layers with distinct resistivity and depth is theoretically calculated by using (2). Therefore, soil parameters, $h_i\,(i=1,2,\cdots,N-1)$ and $\rho_i\,(i=1,2,\cdots,N)$, are going to be estimated in the process of making the theoretical apparent resistivity as close as possible to the measured apparent resistivity. The most satisfied soil parameters can be determined by minimizing the difference between the measured and theoretical apparent resistivity.

3 Estimating Soil Parameters Using the Kernel Function

3.1 Determining the Kernel Function

As known in (3) and (4), the kernel function $f(\lambda)$ is the function of soil parameter (number of soil layers, soil resistivity and depth). In the general optimization methods for estimating soil parameters, the kernel function $f(\lambda)$ is first calculated with any guessed initial values of soil parameters and the theoretical apparent resistivity, which is calculated with the kernel function, is compared with the measured apparent resistivity for various electrode spans. Therefore, the kernel function $f(\lambda)$ has to be calculated in each iterate of optimization algorithms until the most satisfied soil parameter is determined. This procedure consumes huge computational time in optimization iterating as well as calculating the infinite integral in (2).

J. Zou[2] uses the so-called "two-stage method" in which the kernel function $f(\lambda)$ is calculated just once in the first stage and soil parameters are estimated using a general optimization method in the second stage. He introduces the method to obtain the kernel function $f(\lambda)$ without knowing soil parameters. He uses the measured apparent resistivity of various electrode spans to obtain the kernel function $f(\lambda)$. Therefore the obtained the kernel function $f(\lambda)$ can be assumed to contain the true values of soil parameters. He uses the following equation (5) and (6) to obtain the kernel function $f(\lambda)$.

$$f(\lambda)=\lim_{n\to\infty}\sum_{t=1}^{n}\beta_t e^{-\lambda k_t}\approx\sum_{t=1}^{n}\beta_t e^{-\lambda k_t} \tag{5}$$

$$\begin{bmatrix}\beta_1\\ \beta_2\\ \vdots\\ \vdots\\ \beta_n\end{bmatrix}=\begin{bmatrix}\frac{1}{\sqrt{a_1{}^2+k_1{}^2}}-\frac{1}{\sqrt{4a_1{}^2+k_1{}^2}} & & \frac{1}{\sqrt{a_1{}^2+k_n{}^2}}-\frac{1}{\sqrt{4a_1{}^2+k_n{}^2}}\\ \vdots & \ddots & \vdots\\ \vdots & \ddots & \vdots\\ \frac{1}{\sqrt{a_I{}^2+k_1{}^2}}-\frac{1}{\sqrt{4a_I{}^2+k_1{}^2}} & & \frac{1}{\sqrt{a_I{}^2+k_n{}^2}}-\frac{1}{\sqrt{4a_I{}^2+k_n{}^2}}\end{bmatrix}^{-1}\begin{bmatrix}\frac{1}{2a_1}\left(\frac{\rho_1^m}{\rho_1}-1\right)\\ \vdots\\ \vdots\\ \frac{1}{2a_I}\left(\frac{\rho_I^m}{\rho_1}-1\right)\end{bmatrix} \tag{6}$$

where ρ_i^m ($i=1\cdots I$) is the measured apparent resistivity with the test electrode span a_i ($i=1\cdots I$) of the Wenner configuration method and k_t is the t th decay constant corresponding to the t th sampling of integral.

3.2 Estimating Soil Parameters by Analyzing the Kernel Function

J. Zou[2] uses a general optimization method for estimating soil parameters in the second stage of his two stage method. In this paper, the analytical method has been proposed for estimating soil parameters. With this method, soil parameters can be obtained by analyzing the characteristics of the kernel function.

Inspected in (3) and (4), the kernel function has the following characteristics:

i) α_i $(i=1,2,\cdots,N-1)$ converges to 1 as λ increases.

ii) K_i $(i=1,2,\cdots,N-1)$ converges to the constant of $\frac{\rho_{i+1}-\rho_i}{\rho_{i+1}+\rho_i}$ as λ increases.

iii) The kernel function converges to 0 as λ increases.

Using the above characteristics of the kernel function, soil parameters can be inverted through the following procedures:

1) Let let $i=1$, $\alpha_i(\lambda)=f(\lambda)+1$.

2) If $N>2$, go to step 3), otherwise go to step 6).

3) Obtain $K_i(\lambda)$, h_i using $\alpha_i(\lambda)$.

4) Obtain $\alpha_{i+1}(\lambda)$ using $K_i(\lambda)$.

5) If $i<N-1$, let $i=i+1$, go to step 3), otherwise go to step 6).

6) Obtain ρ_{i+1} using $K_i(\lambda)$ and ρ_1 for $i=1,....N$.

In the step 3) of the above algorithm, $K_i(\lambda)$ and h_i can be obtained from $\alpha_i(\lambda)$ through the following procedures

i) Let $H_i=h_{ini}$.

ii) Evaluate $K_i(\lambda)=\frac{\alpha_i(\lambda)-1}{\alpha_i(\lambda)+1}e^{2\lambda H_i}$.

iii) If $K_i(\lambda)$ converges to constant as λ increases, go to step v), otherwise go to step iv).

iv) Adjust H_i using the following equation (7). As seen in the following equati on (7), if $H_i \neq h_i$, then $K_i(\lambda)$ increases or decreases exponentially as λ in creases.

$$K_i(\lambda) = \frac{\frac{2K_i(\lambda)e^{-2\lambda h_i}}{1-K_i(\lambda)e^{-2\lambda h_i}}}{\frac{2K_i(\lambda)e^{-2\lambda h_i}}{1-K_i(\lambda)e^{-2\lambda h_i}}+2} e^{2\lambda H_i} = K_i(\lambda)e^{-2\lambda(h_i - H_i)}. \tag{7}$$

go to step ii).

v) $h_i = H_i$

$$K_i(\lambda) = \frac{\alpha_i(\lambda)-1}{\alpha_i(\lambda)+1} e^{2\lambda h_i}. \tag{8}$$

In the step 4) of the above algorithm, $\alpha_{i+1}(\lambda)$ can be obtained from $K_i(\lambda)$ using (4) as follows

$$\alpha_{i+1}(\lambda) = -\frac{\rho_i}{\rho_{i+1}} \frac{k_i(\lambda)+1}{k_i(\lambda)-1}. \tag{9}$$

4 Numerical Example

If the structure parameters of soil are known, one can generate the apparent resistivity data for different electrode span a by (2). Then these generated data can be used to check the proposed analysis method. Fig. 2 shows the apparent soil resistivity curve calculated by (2). The corresponding the structure parameters are shown in Table 1.

The generated data with eleven different electrode spans in Table 2 are shown in Fig. 2 as marked ‘o’ s. Fig. 3 shows that the generated kernel function $f(\lambda)$ with the

Table 1. Parameters of a three-layer earth structure

Layer No.	Resistivity($\Omega \cdot m$)	Thickness(m)
1	122.96	2.06
2	26.37	10.02
3	284.44	∞

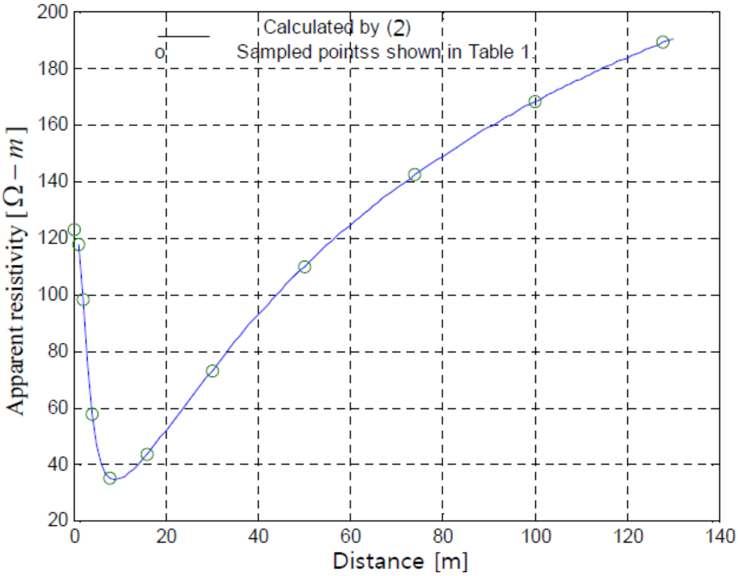


Fig. 2. Apparent resistivity curve of a three-layer earth structure

Table 2. The generated earth apparent resistivity data

No	**a**(m)	ρ($\Omega \cdot m$)	**No**	**a**(m)	ρ($\Omega \cdot m$)
1	0.2	122.9	7	30	80.4
2	1	117.9	8	50	114.6
3	2	98.0	9	74	146.2
4	4	58.3	10	100	172.6
5	8	37.9	11	128	194.4
6	16	50.8			

data in Table 2, which is the J. Zou's method, is almost same as the one which is calculated with the known soil parameters in Table 1. Therefore, the kernel function $f(\lambda)$ generated with the measured apparent resistivity data can be used for the calculated one with the known soil parameters.

When estimating soil parameter with the proposed method, the first step is to guess the number of layers from the apparent resistivity curve. One who has some experience in this area can guess easily that the eleven data marked 'o' in Fig. 2 are

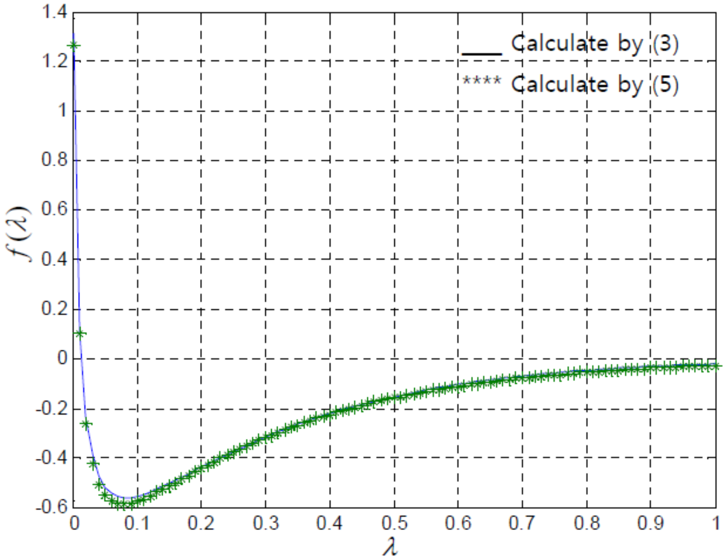


Fig. 3. The kernel function $f(\lambda)$ of a three-layer earth structure

from the 3-layer earth structure. The second step is to obtain the kernel function $f(\lambda)$ with the measured apparent resistivity data by using J. Zou's method. The final step is to estimate soil parameter by analyzing the kernel function. Because the earth structure is guessed as 3-layer from the step one, the soil parameter to be estimated are h_1, h_2, ρ_1, ρ_2 and ρ_3.

Let's use the proposed method in the previous section for estimating soil parameter.

1) $\alpha_1(\lambda)$ can be obtained from the kernel function $f(\lambda)$.
2) Because N=3, go to step 3)
3) h_1 is found as 2.083 for satisfying $K_1(\lambda) = \dfrac{\alpha_1(\lambda)-1}{\alpha_1(\lambda)+1} e^{2\lambda h_1}$ converge to the constant(-0.6468) as λ increases. At the same time, $K_1(\lambda)$ is fixed with h_1.
4) $\alpha_2(\lambda)$ can be obtained using $\alpha_{i+1}(\lambda) = -\dfrac{\rho_i}{\rho_{i+1}} \dfrac{k_i(\lambda)+1}{k_i(\lambda)-1}$ in (9). Because $K_1(\lambda)$ is obtained in step 3) and $\dfrac{\rho_1}{\rho_2}$ can be obtained from $K_1(\lambda)$. The constant (-0.6468), to which $K_1(\lambda)$ converges, is the ratio of $\left(\rho_2 - \rho_1 \middle/ \rho_2 + \rho_1\right)$ as seen in the Fig. 4.

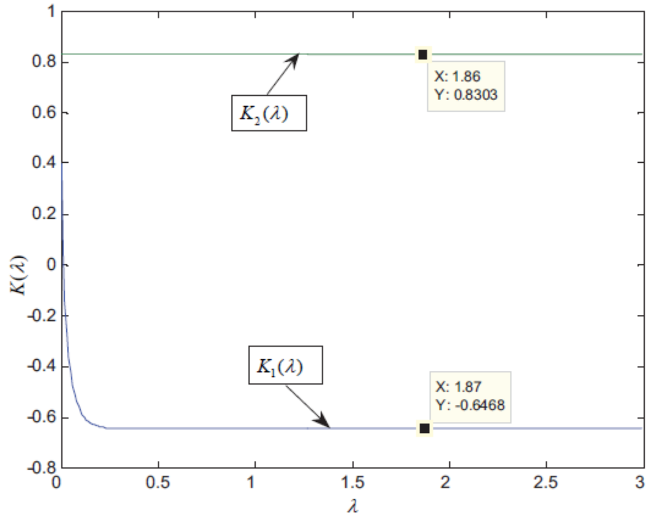


Fig. 4. The obtained $K_i(\lambda)$ $(i=1,2)$ using the proposed method

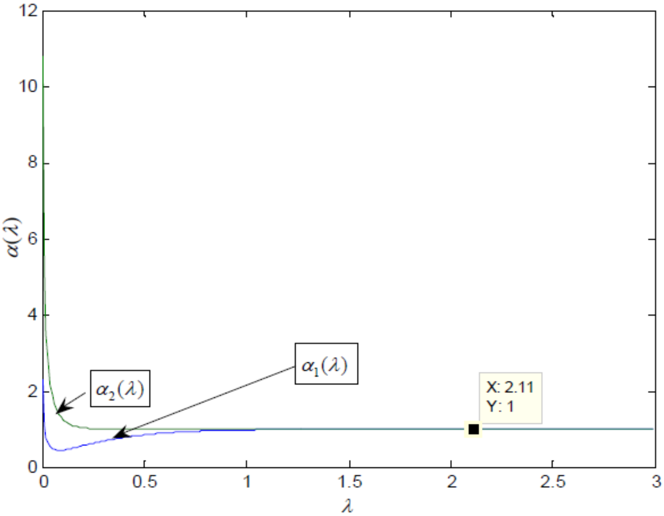


Fig. 5. The obtained $\alpha_i(\lambda)$ $(i=1,2)$ using the proposed method

5) Go to step 3), h_2 is found as 9.989, $K_2(\lambda)$ converges to the constant (0.8303) which is the ratio of $\left(\rho_3-\rho_2 \big/ \rho_3+\rho_2\right)$. Assumed ρ_1 is known from the beginning, then all of the soil parameters (h_1, h_2, ρ_1, ρ_2 and ρ_3) are obtained. The obtained value 2.083 and 9.989 for h_1 and h_2 are in good agreement with the soil depth data in

Table 1. And the obtained ratios of $\frac{\rho_1}{\rho_2}$ and $\frac{\rho_2}{\rho_3}$ are exactly in agreement with the soil resistivity data in Table 1. The obtained $K_i(\lambda)$ $(i=1,2)$ and $\alpha_i(\lambda)$ $(i=1,2)$ from the above procedure are shown in Fig. 4 and Fig. 5 respectively.

5 Conclusion

Generally various optimization algorithms are used for estimating the soil parameters of multilayer earth structure. Those methods require the difficult derivatives of the optimized expression and huge computational time. The estimating procedure in this paper is composed of two stages: the kernel function $f(\lambda)$ is solved using J. Zou's method in the first stage and the proposed method is used for obtaining soil parameters in the second stage. As seen in numerical example, the proposed method simplifies the problem by not requiring the derivatives of optimization expression and the estimated soil parameters are in good agreement with the given earth structure.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0077743).

References

1. Zhang, B., Cui, X., Li, L., He, J.L.: Parameter Estimation of Horizontal Multilayer Earth by Complex Image Method. IEEE Trans. on Power Delivery 20, 1394–1401 (2005)
2. Zou, J., He, J.L., Zeng, R., Sun, W.M., Chen, S.M.: Two-Stage Algorithm for Inverting structure Parameters of the Horizontal Multilayer Soil. IEEE Trans. on Magnetics 40, 1136–1139 (2004)
3. Dawalibi, F.: Earth Resistivity Measurement Interpretation Techniques. IEEE Trans. on Power Apparatus Systems 103, 374–382 (1984)
4. Takahashi, T., Kawase, T.: Analysis of Apparent Resistivity in a Multi-layer Earth Structure. IEEE Trans. on Power Delivery 5, 604–612 (1990)
5. Dawalibi, F.P.: Electromagnetic Fields generated by overhead and buried short conductors. IEEE Trans. on Power Delivery 1, 105–119 (1986)
6. Zhang, B., Zhao, Z., Cui, X.: Diagnosis of Breaks in Substation's Grounding Grid by Using Electromagnetic Method. IEEE Trans. on Magnetics 38, 473–476 (2002)

A Genetic Algorithm for Efficient Delivery Vehicle Operation Planning Considering Traffic Conditions

Yoong-Seok Yoo and Jae-Yearn Kim

Department of Industrial Engineering, Hanyang University,
Sungdong-gu, Seoul 133-791, Korea
ysyoo@hanyang.ac.kr,
jyk@hanyang.ac.kr

Abstract. To ensure customer satisfaction, companies must deliver their product safely and within a fixed time. However, it is difficult to determine an inexpensive delivery route when given a number of options. Therefore, an efficient vehicle delivery plan is necessary. Until now, studies of vehicle routes have generally focused on determining the shortest distance. However, vehicle capacity and traffic conditions are also important constraints. We propose using a modified genetic algorithm by considering traffic conditions as the most important constraint to establish an efficient delivery policy for companies. Our algorithm was tested for fourteen problems, and it showed efficient results.

Keywords: Genetic Algorithm, Delivery Vehicle Operation Planning.

1 Introduction

1.1 Vehicle Routing Problem

Although it is important for companies to produce high-quality products, with the recent development in the Internet environment and the expansion of electronic commerce, many companies are faced with the problem of satisfying customers by delivering the product to the correct place and on time, as promised. These considerations are even greater for third-party logistics companies.

Logistics companies must satisfy the delivery demands of customers. If a company does not meet these demands, customers will feel that the product is less reliable, and this affects the manufacturing company over a long time. Therefore, the date of delivery affects not just the logistics company but also the manufacturer.

Many practical constraints affect delivery, including the number of vehicles the company owns, the operating time for each vehicle, road conditions, and uncertain customer demand. Short-term changes in traffic conditions are one of the important constraints that affect timely product delivery.

Typically, previous studies on the vehicle routing problem (VRP) have disregarded this constraint and simply examined the shortest route and minimum vehicle operation to minimize the operating costs. In practice, such solutions have often been applied by companies. However, it is now possible improve the efficiency of vehicle operation by

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 119–129, 2010.

considering traffic conditions using real-time road information obtained via wireless Internet. The traffic situation is a serious constraint that affects the delivery policy.

The VRP can be solved using optimal algorithms such as linear programming (LP) and the branch and bound (B&B) method or heuristic algorithms such as tabu search (TS), simulated annealing (SA), and genetic algorithms (GA). However, the calculation time increases markedly with the size of the delivery region. Therefore, heuristic algorithms that can calculate rapidly the local optimum have recently been developed.

Dantzig and Ramser [6] showed that the VRP is NP-hard; for example, when the number of branches is 15, the number of routes is 653,837,184,000. Christofides and Eilon [4] developed a B&B method using a Lagrangian relaxation technique to include the constraint of the maximum vehicle operating distance in the basic VRP. Nevertheless, calculation time increases markedly with the number of branches in the B&B method. Clarke and Wright [5] proposed a method that considers the cost of using one vehicle to deliver products to two customers, rather than using two vehicles. Gendreau et al. [9] developed a solution that uses the tabu search concept. The strength of this method is that it produces an excellent solution but, again, it has the disadvantage of a long calculation time. Lee and Kim [11] were the first to apply a parallel GA to a job-scheduling problem for a single machine. However, they used a binary form of a gene and displayed it using a one-dimensional array. Cheng and Gen [2] subsequently applied a parallel GA to a job-scheduling problem using identical parallel machine systems. They used a floating gene, but still used a one-dimensional array in their genetic representation. They used a distinct symbol to distinguish between each machine.

1.2 Determining Constraints

A company must consider various constraints in the vehicle operation problem, such as the number of vehicles, avoiding duplicate visits, vehicle capacity, completing deliveries within a fixed time, delays in delivery due to traffic conditions, rest for the drivers, and canceled orders. Since the first four constraints are often readily quantified, they have been examined in many studies. Some constraints can be overcome simply by increasing the number of vehicles, while others are unmanageable.

Among the four constraints considered basic to the VRP, changes in traffic conditions are often the bottleneck constraint. Therefore, we considered this factor to be the most important bottleneck constraint.

First, we considered changes in traffic conditions in real time. We weighted the delivery distance between the supplier and customers according to the traffic conditions that were classified as normal traffic, slow, jammed, and accident. The Normal traffic condition indicates vehicles running at normal speed. The slow indicates vehicles traveling more slowly than in the normal state; in the case a traffic jam, the road conditions bad. An accident was defined as a traffic accident or road construction site that blocks the use of a road temporarily. A logistics company can use these four categories and apply different weights to each category (Table 1).

Table 1. Weights and occurrence ratios of traffic conditions

Condition	Normal flow	Slow flow	Traffic jam	Accident
Velocity (km/h)	>60	30~59	10~29	<10
Weight	1.0	1.5	1.8	2.0
Ratio	50%	30%	15%	5%

Much data must be analyzed to determine the weights accurately. Initially, we arbitrarily used 1.0 as the weight for normal traffic and assumed weights of 1.5, 1.8, and 2.0 for the slow flow, traffic jam, and accident conditions. In addition, we assumed that the occurrence ratio between the four conditions was 5:3:1.5:0.5.

To calculate the cost of a route, we multiplied the distance by the weight. If the traffic is serious, the delivery time is increased. Our method searches for alternative routes where the traffic is smooth. According to our method, a vehicle might travel a longer distance but it arrives at the customer's location more quickly than for a shorter route. If no such route is found using one vehicle, an alternative may be to use an additional vehicle. In this manner, timely delivery is available to the customers is achieved.

2 Modified Genetic Algorithm

In Section 2.1, we explain how to develop a two-dimensional array-type gene. In Section 2.2, we develop operators for a two-dimensional array-type gene.

2.1 Two-Dimensional Array Type Gene

We used a two-dimensional array form for the genes, as shown in Fig. 1. Each row of the gene represents a vehicle. Each column identifies the customers that each vehicle visits.

	Customer Number				
Vehicle 1	1	4	7	10	X
Vehicle 2	5	2	12	11	8
Vehicle 3	9	6	3	X	X

Fig. 1. Two-dimensional array

In the above example, three vehicles, one head office, and 12 customers are considered. The delivery route of the first vehicle is 0 →1 → 4 → 7 → 10 → 0, where 0 represents the head office. A route must satisfy the maximum capacity of the vehicle and the largest operating distance constraint. The genetic factor denoted by X indicates that a customer could not be included because one of the two constraints was not satisfied.

According to Michalewicz [13], genetic algorithms consider the following five elements:

- A genetic representation of something that enables achieving a solution.
- A way to generate an initial population.
- A fitness function that evaluates the solution.
- A genetic operator that changes the make-up of the children.
- The values of the various parameters used.

We now explain how the five elements were modified in the proposed algorithm.

First, a traditional GA uses a binary array to represent a gene. If there are N customers and K vehicles, an $N \times N \times K$ dimensional array is used. As the number of branches increases, the memory requirements and calculation time increase, eventually making it almost impossible to solve the problem. Our model uses a real-type (floating) array in the algorithm, so an $N \times K$ dimensional array is needed. In general, the number of vehicles is less than the number of customers. Our method saves memory and reduces the calculation time.

Our GA differs from the existing ones. It uses a real number (not binary) for genes. In addition, our algorithm uses the two-dimensional array form for the genes. A gene expressed using real numbers better represents actual situations. Moreover, the computation time is reduced. This is because a conversion process is not necessary in our algorithm. Moreover, the memory storage is reduced even when the number of delivery locations is large. Therefore, the advantage of this method is that it efficiently calculates a solution to a large problem.

Second, a traditional genetic algorithm uses random sampling to generate the initial population. This can duplicate and/or omit customers. Therefore, the initial population may result in infeasible solutions and/or unnecessary calculations, decreasing the execution speed of the algorithm. Our algorithm does not duplicate customers in generating the initial population, thus eliminating unrealizable solutions at the beginning of the evolutionary process. Therefore, we can find the optimal solution quickly.

Third, since the traditional GA uses a gene in a binary array form, it is necessary to convert it into a real number to evaluate the fitness of a solution. This process may generate errors and it increases the calculation time. We use a “real” type gene to overcome this problem.

Fourth, the traditional GA cannot use traditional genetic operators. The PMX, CX, and OX operators were developed to overcome these problems [2]. However, these operators are only suitable for genes with a one-dimensional array form. Since we use a two-dimensional array, we cannot use the existing genetic operators. Therefore, we developed new genetic operators suitable for two-dimensional arrays, as explained in detail below.

Fifth, the parameters used in a GA produce marked differences in the calculation time and quality of the solution. The optimal values have to be determined by performing repeated experiments by taking various levels into consideration. We also determined the parameters suitable for our algorithm in experiments at various levels.

2.2 Evaluating Fitness of Solution

The fitness of a solution is evaluated using the following formula.

$$Minimize \sum_{K=1}^{M}\sum_{i=0}^{N}\sum_{j=0}^{N} c_{ijk}\ x_{ijk} \tag{1}$$

Where C_{ijk} is the cost of moving vehicle k from customer j to i and X_{ijk} is an integer such that if vehicle k is allocated to customers i and j, then $X_{ijk} = 1$; otherwise $X_{ijk} = 0$. This function ensures that the demands of all customers are satisfied while minimizing the vehicle operating costs.

2.3 Developing Genetic Algorithm Operators

The crossover operator interchanges genes between two parents randomly and generates new genes in the offspring. This can produce solutions in the offspring that are better than those in the parents. Cheng and Gen [2] used the PMX, OX, and CX operators, which include a crossover operator that can be applied to the VRP. However, these operators can only be applied to a gene represented by a one-dimensional array. Therefore, we modified the OX operator so that it can be applied to a gene represented by a two-dimensional array. The modified OX operator is executed in Fig. 2.

Step 1: Randomly select two vehicles (point 1, point 2) from two parents at a crossover rate.

Step 2: Remove customers of the selected vehicle in the gene of the second parent.

Step 3: Exchange the customers of the vehicles of the first parent that were not selected in Step 1 with those remaining in the second parent.

In this manner, child 1 and child 2 are generated. No customers are duplicated or omitted in child 1 and child 2.

Our modified OX crossover operator carries out crossover operations on vehicle units. Therefore, children obtain copies of routes of the selected vehicle that satisfy existing constraints. Thus, the positive nature of the parents' solution is passed to the following generation.

Parent 1		1	2	3	4	5
	Point 1 →	6	7	8	X	X
		9	10	11	12	X
Parent 2	Point 2 →	1	4	7	10	X
		2	5	8	11	12
		3	6	9	X	X
Child 1		1	4	10	2	5
		6	7	8	X	X
		11	12	3	9	X
Child 2		1	4	7	10	X
		2	3	5	6	8
		9	11	12	X	X

Fig. 2. Example of modified OX crossover operator

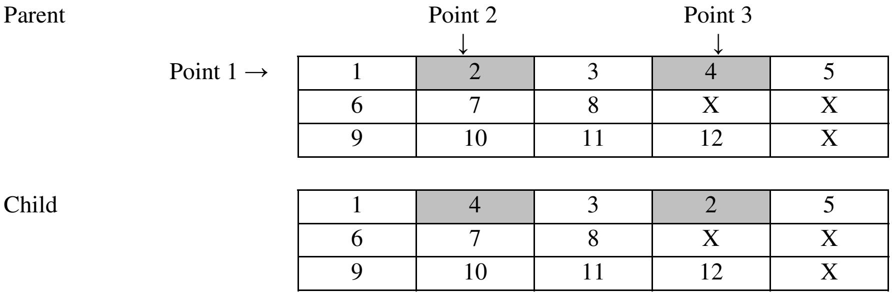
We use two mutation operators that modify reciprocal exchange mutations [2]. A mutation operator generates a solution that differs from the existing solutions in the parents so that the genetic algorithm does not get stuck at a local optimum solution.

Our mutation operator is subdivided into internal and external mutation operators. Our genes are represented by two-dimensional arrays. An internal mutation occurs within one row (vehicle), while an external mutation occurs between different rows (vehicles). In other words, an internal mutation is a mutation that affects one vehicle, and an external mutation is a mutation affecting different vehicles.

An internal mutation is generated as follows:

Step 1: Select one vehicle (point 1) at random after selecting the parent at mutation rate.
Step 2: Select two customers (point 2, point 3) in the selected vehicle.
Step 3: Interchange the two selected customers.

Fig. 3 shows an example of an internal mutation.



Parent

1	2	3	4	5
6	7	8	X	X
9	10	11	12	X

Child

1	4	3	2	5
6	7	8	X	X
9	10	11	12	X

Fig. 3. Example of internal mutation

An external mutation is generated as follows:

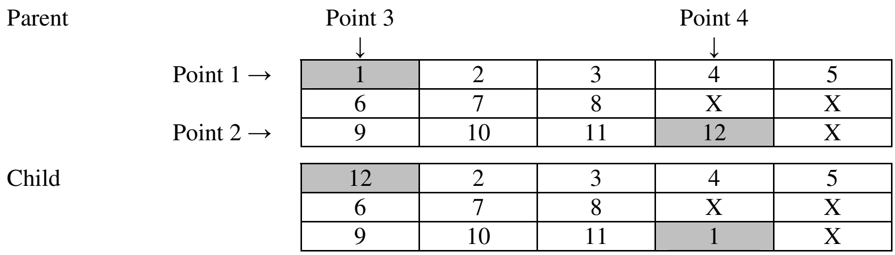
Step 1: Select a pair of vehicles (point 1, point 2) at random after selecting the parent at the mutation rate.
Step 2: Select one customer in every vehicle (point 3, point 4).
Step 3: Interchange the two selected customers.

Fig. 4 shows an example of an external mutation.

As mentioned above, the two mutation operators prevent the solution from getting stuck at a local optimum.

We use a reversion operator to facilitate escape from a local optimum. One disadvantage of the genetic algorithm is the slow speed of convergence on the solution. The reversion operator overcomes this disadvantage as follows.

Step 1: Select a pair of vehicles (point 1) at random after selecting the parents at reversion rate.
Step 2: Select one customer (point 2) in the selected vehicle.
Step 3: Change the order of customers so that it starts with the one in the middle.



Parent

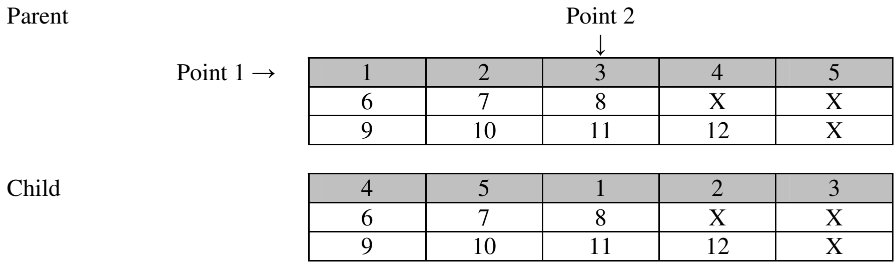
1	2	3	4	5
6	7	8	X	X
9	10	11	12	X

Child

12	2	3	4	5
6	7	8	X	X
9	10	11	1	X

Fig. 4. Example of external mutation

Fig. 5 shows an example of an external mutation.



Parent

1	2	3	4	5
6	7	8	X	X
9	10	11	12	X

Child

4	5	1	2	3
6	7	8	X	X
9	10	11	12	X

Fig. 5. Example of modified reversion operator

Our genetic operator does not generate duplicate customers. Nevertheless, it can violate the vehicle capacity or operating distance constraints. Therefore, the solution must be revised after carrying out each operation. First, we look for the solutions that violated a given constraint and exchange these solutions for new solutions.

3 Numerical Experiment

Our algorithm was used to solve 14 vehicle routing problems that can be downloaded from the OR-library [14]. These VRPs are widely used as benchmarks. The problems considered in the experiment are summarized in Table 2. Our algorithm was coded in the C language. Our experiments were performed on a Sony laptop computer equipped with 1 GB of memory and a Pentium-M processor running at 2.13 GHz running the Windows XP Professional operating system.

These problems have 50–199 customers. We can classify these problems in two ways: vrpnc1~vrpnc5, vrpnc11, and vrpnc12 are only constrained by the vehicle capacity, while the others consider vehicle capacity, maximum route time, and drop time. The number of vehicles, range of demand, and vehicle capacity also differ among the problems. Therefore, we can evaluate the effect of our algorithm in various environments.

Table 2. Problem summary

No. of Problems	No. of Customers	Vehicle Capacity	No. of Vehicles	Maximum Route Time	Drop Time
vrpnc1	50	160	5	∞	0
vrpnc2	75	140	10	∞	0
vrpnc3	100	200	8	∞	0
vrpnc4	150	200	12	∞	0
vrpnc5	199	200	17	∞	0
vrpnc6	50	160	6	200	10
vrpnc7	75	140	11	160	10
vrpnc8	100	200	9	230	10
vrpnc9	150	200	14	200	10
vrpnc10	199	200	18	200	10
vrpnc11	120	200	7	∞	0
vrpnc12	100	200	10	∞	0
vrpnc13	120	200	11	720	50
vrpnc14	100	200	11	1040	90

3.1 Parameter Optimization

With the genetic algorithm, the speed of convergence differs according to the parameters used. Our GA uses new genetic operators, so new parameters have to be determined. Because we use new genetic operators, new parameters are chosen accordingly. Our GA uses four parameters: size of the initial population, crossover rate, mutation rate, and reversion rate.

We examined one problem (vrpnc1) and selected two preliminary levels, which we then applied to all problems for determining the appropriate level. The number of generations was fixed at 100,000. We terminated the algorithm whenever there was no improvement for 20,000 generations. The size of the initial population was selected to be 50. The crossover rate was selected to be 0.8, and the mutation rate was selected to be 0.2. The reverse rate was selected to be 0.3.

3.2 Numerical Example

We demonstrate the effectiveness of our algorithm by considering a small test problem. This problem was made by Heyes [10]; it considers six customers. The maximum vehicle capacity is 3, and the demand for each customer is 1. The company owns two vehicles. The distance between customers and the traffic conditions are shown in Table 3. The first number denotes the distance, while the second denotes the traffic condition.

We found the best and worst routes. The best solution considered road conditions, but the worst did not do so. The solutions for the problem used two vehicles and the delivery schedule shown in Table 4.

The best solution's total operating cost was 129.0. However, the worst solution's total operating cost was 296.5. Therefore, the gap between the two solutions was 167.5. Therefore, if we can obtain information about the road conditions, we can reduce the delivery time and total cost.

Table 3. Distance between customers and traffic conditions (distance/traffic condition)

Customer	0	1	2	3	4	5	6
0	-	10 / 1.0	20 / 1.0	25 / 1.5	25 / 1.0	20 / 1.5	10 / 1.5
1	10 / 1.0	-	12 / 1.5	20 / 1.0	25 / 1.0	30 / 2.0	20 / 1.0
2	20 / 1.5	12 / 1.8	-	10 / 1.8	11 / 1.0	22 / 1.0	30 / 1.8
3	25 / 1.5	20 / 1.0	10 / 1.0	-	2 / 1.5	11 / 1.8	25 / 1.0
4	25 / 1.8	25 / 1.0	11 / 1.5	2 / 1.0	-	10 / 1.5	20 / 1.0
5	20 / 1.0	30 / 1.5	22 / 1.0	11 / 1.5	10 / 1.0	-	12 / 1.5
6	10 / 1.5	20 / 1.0	25 / 1.8	25 / 1.0	20 / 2.0	12 / 1.0	-

Table 4. Delivery schedule for the small test problem

	Vehicle	Delivery route	Operating cost	Total cost
Best Solution	1	0 → 1 → 6 → 5 → 0	62.0	129.0
	2	0 → 4 → 3 → 2 → 0	67.0	
Worst Solution	1	0 → 2 → 6 → 4 → 0	159.0	296.5
	2	0 → 3 → 1 → 5 → 0	137.5	

Table 5. Experimental results considering traffic conditions

No. of Problems	No. of Customers	Avg. Best Solution	Avg. Worst Solution	Gap	% Gap
vrpnc1	50	547.8	659.7	111.9	20.427
vrpnc2	75	872.6	1032.1	159.5	18.279
vrpnc3	100	882.2	1054.9	172.7	19.576
vrpnc4	150	1097.3	1288.5	191.2	17.425
vrpnc5	199	1429.8	1893.9	464.1	32.459
vrpnc6	50	592.5	717.6	125.1	21.114
vrpnc7	75	973.6	1275.3	301.7	30.988
vrpnc8	100	902.1	1082.7	180.6	20.020
vrpnc9	150	1232.7	1633.2	400.5	32.490
vrpnc10	199	1537.8	1827.6	289.8	18.845
vrpnc11	120	1098.1	1329.9	231.8	21.109
vrpnc12	100	902.9	1102.2	199.3	22.073
vrpnc13	120	1601.1	1928.4	327.3	20.442
vrpnc14	100	919.5	1029.3	109.8	11.941
			Average	**233.2**	**21.942**

3.3 Experimental Results

We solved each problem 20 times. In each problem, the traffic conditions were generated at random. Table 5 shows the obtained results.

The average gap between the best and worst solutions was 233.2, which is an improvement of 21.942%. Therefore, if a company uses traffic condition information, it can reduce the operation cost and satisfy its customers.

4 Conclusion

We used a GA to determine efficient and realistic delivery vehicle operation plans. The number of available vehicles or their capacity no longer constitutes serious problems for logistics companies. Nowadays, traffic conditions are the main problem affecting a company's ability to deliver products to customers on time. If companies cannot overcome this problem, they lose customers. Therefore, we considered traffic conditions as the most important constraint. To determine more efficient vehicle routes, our delivery plans incorporate traffic information.

We classified the traffic conditions as normal, slow, jammed, and accident. We weighted each traffic condition according to its frequency. Our solutions focused on the delivery time rather than on the distances to the customers.

Our GA uses a two-dimensional array, with each row representing a different vehicle and each column representing customers. Therefore, it is easier to represent a real system using our GA than the conventional GAs. Our GA could be used to find reasonable solutions rapidly.

Our algorithm was tested for 14 problems. The best solution had an operational cost that was lower than the worst solution by 233.2 on an average, which indicated an improvement of 21.942%. Therefore, if a company uses information regarding traffic condition, it can reduce the operational costs and satisfy customers.

In future, we hope to quantify traffic conditions and collect data regarding traffic conditions in real time.

References

1. Baker, B.M., Ayechew, M.A.: A Genetic Algorithm for the Vehicle Routing Problem. Computers & Operations Research 30, 787–800 (2003)
2. Cheng, R., Gen, M.: Genetic Algorithm and Engineering Design. John Wiley & Sons, New York (1996)
3. Prins, C.: A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem. Computers & Operations Research 31, 1985–2002 (2004)
4. Christofides, N., Eilon, S.: An Algorithm for the Vehicle Dispatching Problem. Operational Research Quarterly 20(3), 309–318 (1969)
5. Clarke, G., Wright, J.: Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. Operations Research 11(4), 568–581 (1963)
6. Dantzig, G.B., Ramser, J.H.: The Truck Dispatching Problem. Management Science 6(1), 80–91 (1959)
7. Dolce, J.: Fleet Management. McGraw-Hill, New York (1984)

8. Gaskell, T.J.: Bases for Vehicle Fleet Scheduling. Operational Research Quarterly 18(3), 281–295 (1967)
9. Gendreau, M., Hertz, A., Laporte, G.: A Tabu Search Heuristic for the Vehicle Routing Problem. Management Science 40(10), 1276–1290 (1994)
10. Hayes, R.L.: The Delivery Problem. Carnegie Institute of Technology, Graduate School of Industrial Administration, Pittsburgh, Report No. MSR 106 (1967)
11. Lee, C., Kim, S.: Parallel Genetic Algorithm for the Tardiness Job Scheduling Problem with General Penalty Weights. International Journal of Computers and Industrial Engineering 28, 231–243 (1995)
12. Toth, P., Vigo, D.: The Vehicle Routing Problem. Society for Industrial and Applied Mathematics, Philadelphia (2002)
13. Michalewicz, Z.: Genetic Algorithm + Data Structure = Evolution Programs. Springer, Heidelberg (1996)
14. The OR-Library, `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`

On the Possibilities of Multi-core Processor Use for Real-Time Forecast of Dangerous Convective Phenomena

Nikita Raba[1] and Elena Stankova[1,2]

[1] Saint-Petersburg State University
199034 St.-Petersburg, Russia
[2] Institute for High Performance Computing and Integrated Systems,
199397 St.-Petersburg, Russia
no13@inbox.ru, lena@csa.ru

Abstract. We discuss the possibilities of use of the new generation of desktops for solution of one of the most important problems of weather forecasting: real-time prediction of thunderstorms, hails and rain storms. The phenomena are associated with development of intensive convection and are considered as the most dangerous weather conditions. The most perspective way of the phenomena forecast is computer modeling. Small dimensional models (1 - D and 1.5 - D) are the only available to be effectively use in local weather centers and airports for real-time forecasting. We have developed one of such models: 1.5 - D convective cloud model with the detailed description of microphysical processes and have investigated the possibilities of its parallelization on multi-core processors with the different number of cores. The results of the investigations have shown that speed up of cloud evolution calculation can reached the value of 3 if 4 parallelization threads are used.

Keywords: multi-core processors, parallelization, thread, numerical model, real-time weather forecast, convective cloud.

1 Introduction

Climate and weather forecast is among the so called grand-challenge scientific problems, which need for their solution high-performance computer facilities. All the main weather centers in the world are equipped with powerful clusters and supercomputers. But one should take into account that weather forecast in not only the prediction of wind and pressure fields which are the output of the so called regional models and general circulation models of atmosphere, but also the prediction of local dangerous convective phenomena, such as thunderstorms, hails and rain storms. Forecast of rain, hail and thunderstorm is usually provided in rather small weather centers and airports which have modest financial resources and are not able to buy expensive supercomputers or even clusters. Ordinary desktops are the only computational resources that are available. The problem is even more complicated as the forecast should be real-time and it should take no more than one an hour to provide it. As a consequence experts of such local centers have to provide forecast with the help of simple methods and models. Up to now forecast of the dangerous

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 130–138, 2010.

convective phenomena in the airports of CIS (Commonwealth of Independent States) countries is provided with the help of semi-empirical methods and very simple 1-D stationary cloud models. It is evident that requirement of real-time forecast in combination with modest computational resources will not allow using elaborated 2 - D and 3 - D models in such centers. But appearance of desktops with multi-core processors open the possibility of applying elaborated 1-D cloud models with detailed description of microphysical processes. The only requirement is proper use of multi-core processor facilities by means of parallelization.

We have developed 1.5-D convective cloud model with detailed description of microphysical processes and have investigated possibilities of its effective use for real-time forecast of cloud parameters. Calculations have been provided with the help of multi-core processors of different types and different core numbers.

The so called space parallelization in conjunction with the multi-thread technology has been used. The results have shown that speed up of cloud evolution calculation can reached the value of 3 if 4 parallelization threads are used.

2 Model Description

In the model the region of convective flow is represented by two concentric cylinders [1]. The inner cylinder (with constant radius a) corresponds to the updraft flow region (cloudy region) and the outer cylinder (with constant radius b) – to the surrounding downdraft flow region (cloudless) (Fig.1)

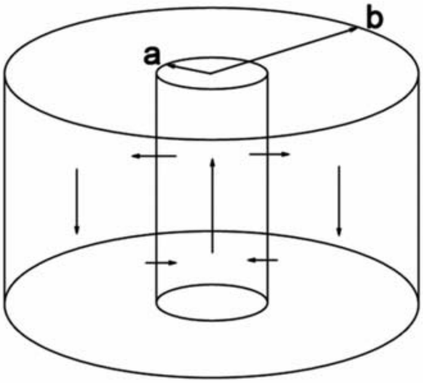


Fig. 1. The scheme of up and down flows

The model is 1.5-dimensional with the detailed description of warm (i.e. without the ice phase) microphysical processes. The term 1.5 – dimensional means the following: though all cloud variables are represented with mean values averaged over the horizontal cross section of the cloud, fluxes in and out of the inner cylinder borders are taken into account.

The ratio of the area of cross section of inner cylinder to the area of cross section of outer ring-shaped cylinder is equal to

$$K_{ab} = a^2 / (b^2 — a^2) \tag{1}$$

In generalized form the equations for vertical velocity, temperature and mixing ratios of water vapour and cloud droplets inside the inner (equation 2) and outer (equation 3) cylinders can be written as follows:

$$\frac{\partial \phi_{in}}{\partial t} = -w_{in}\frac{\partial \phi_{in}}{\partial z} - \frac{2\alpha^2}{a} | w_{in} - w_{out} | (\phi_{in} - \phi_{out}) + \frac{2}{a} U_a (\phi_{in} - \phi_a) + \\ \frac{1}{\rho_{a_0}} \frac{\partial}{\partial z} K_f \frac{\partial \phi_{in}}{\partial z} + F_{\phi_{in}} - A_{\phi_{in}} + G_{\phi_{in}}, \tag{2}$$

$$\frac{\partial \phi_{out}}{\partial t} = -w_{out}\frac{\partial \phi_{out}}{\partial z} - \frac{2\alpha^2}{a} | w_{in} - w_{out} | (\phi_{out} - \phi_{in}) + \frac{2}{a} U_a (\phi_{out} - \phi_a) + \\ \frac{1}{\rho_{a_0}} \frac{\partial}{\partial z} K_f \frac{\partial \phi_{out}}{\partial z} + F_{\phi_{out}} - A_{\phi_{out}} + G_{\phi_{out}}, \tag{3}$$

Where the variables with subscripts 'in' and 'out' relate to the values, averaged over the inner and outer cylinders consequently. ϕ can take the values of vertical velocity *w*, temperature *T*, mixing ration of water vapor Q_v and mixing ratio of cloud droplets in the *i*-th drop-size interval Q_{ci}. *t* and *z* are independent variables (time and height consequently), α is the coefficient for lateral eddy mixing through the periphery of the cloud, U_a is determined by the equation of mass continuity under assumption of incompressibility which is given as $\frac{2u_a}{a} + \frac{1}{\rho_{a_0}} \frac{\partial(\rho_{a_0} w_{in})}{\partial z} = 0$, ρ_{a_0} is density of the atmospheric air, K_f is the turbulent viscosity coefficient.

Concrete form of the terms F_ϕ, A_ϕ, G_ϕ depends upon the meaning of ϕ.

$F_w = \frac{g(T_v - T_{v0})}{T_{v0}} - gQ_c$, where T_v is the virtual temperature, T_{v_0} - is the virtual temperature averaged over the cross sections of the both cylinders, Q_c is the total mixing ratio of cloud drops $Q_c = \sum_{i=1}^{N-1} Q_{ci}$, $F_T, F_{Q_v}, F_{Q_{ci}}$ describe the input of the microphysical process of condensation into the change of temperature and mixing rations of water vapor and cloud droplets in the *i*-th drop-size interval consequently. $A_w = A_{Q_v} = 0$, $A_T = -\gamma_a w / T$, where γ_a is the dry adiabatic lapse rate, $A_{Q_{ci}} = V_{di} \frac{\partial}{\partial z} (Q_{ci})$, where V_{di} - is the value of cloud drop terminal velocity of in the *i*-th drop-size interval, $G_w = 0$, $G_T, G_{Q_v}, G_{Q_{ci}}$ describe the input of the microphysical process of evaporation into the change of temperature and mixing rations of water vapor and cloud droplets in the *i*-th drop-size interval consequently.

The detailed description of the dynamical part of the model is presented in [2].

It is well-known now that in order to predict cloud evolution characteristics properly one must use drop size dependent theories that is to include the equation describing the evolution of the number density function of the cloud drops *f* into the system of cloud equations. Function $f = f(\vec{x}, m, t)$, where m is drop mass, varies in a given space point ($\vec{x}$) due to the processes of advection, sedimentation, turbulent mixing, condensation, nucleation and collection.

For the numerical solution of the equation it is necessary to select discrete points m_i ($i = 0,..., N, m_0 = 0$) along the m axis to define drop size intervals or bins. Then one can replace the stochastic collection equation by the set of equations for M_i - mass fraction in the mass interval defined by:

$$M_i = \int_{m_{i-1/2}}^{m_{i+1/2}} mf(m)dm, \tag{4}$$

$i = 1,..., N - 1$. So an equation for $\frac{\partial M_i}{\partial t}$ can be written in the following form:

$$\frac{\partial M_i}{\partial t} + \nabla \cdot \left[\left(\vec{V} + \vec{V}_{di} \right) M_i \right] = \nabla \cdot (K_f \nabla M_i) + J^n \delta_{i1} + C_i + S_i^+ - S_i^- \tag{5}$$

where $\vec{V}$ is velocity vector, $\vec{V}_d$ is terminal velocity of the drop, K_f is turbulent diffusion coefficient, C_i is rate of condensation (the growth of the drop due to the diffusion of water vapour) of the particle with mass m, J^n is rate of nucleation: rate of formation of the droplet of mass which belongs to the first drop mass interval.

$$\vec{V}_{d_i} = \frac{\int_{m_{i-1/2}}^{m_{i+1/2}} \vec{V}_d(m) mf(m) dm}{\int_{m_{i-1/2}}^{m_{i+1/2}} mf(m)dm} \approx \vec{V}_d(m_i), \tag{6}$$

$$C_i = \int_{m_{i-1/2}}^{m_{i+1/2}} m \frac{\partial \dot{m} f(m)}{\partial m} dm,$$

Terms $S_i^+ - S_i^-$ characterize the process of collection: particle growth due to the collision of the drops with each other.

3 Numerical Scheme of the Model

The method of physical process splitting is used for solution of the system of the equations. Only dynamical processes are taken into account at the first stage. Equations are numerically integrated using a finite difference method. Forward-upstream scheme is used. Vertical velocity is averaged over two grid points (point below is taken if w≥0 or point above if w<0). The final values are obtained on the second stage after completion of the microphysical processes calculation. A time step Δt of 1 sec and a height interval Δz of 200 m are used.

The height of the cylinder is 15 km. The temperature at the ground surface is 298K. The temperature laps rate is 9,8 K/km up to 2 km and is 6,3 K/km from 2 km to 10 km. The temperature is constant above 10 km. The relative humidity is 100% at the ground and decreases with lapse rate of 5%/km up to the top of cylinder. Initial contents of cloud droplets (Q_c) is equal to zero at all levels. Vertical (w) and radial (u_a) velocities and Q_c are assumed to be 0 at the top and at the bottom boundaries of the cylinder. Q_r and Q_i are equal to zero at the top boundary. The initial disturbance of vertical velocity in the inner cylinder below 2 km is given as

$$w = \Delta w \cdot z \cdot (2 - z) \tag{7}$$

where Δw is taken as 1 m/sec. The coefficient for lateral eddy mixing is 0,1. The vertical eddy diffusion coefficient equals to 100 m^2/sec.

Numerical scheme similar to that used in [3] was used for calculation of nucleation and condensation processes and similar to that used in [4] for calculation of collection.

4 Comparison of Microphysical and Dynamical Process Impact into Model Time Calculation

Each cloud model consists of two main blocks: dynamical one, which is responsible for calculation of velocity components and further transport of temperature and bulk characteristics of water vapor and cloud droplets, and microphysical block, which is responsible for calculation of distribution function of cloud droplets.

Dynamical block in 1-D and 1.5-D cloud droplets is rather simple and does not demand essential computational resources. Calculations have shown that it takes only 15 seconds of computer time to obtain full set of dynamical characteristics of model cloud, evaluating during 60 min. The result is quite acceptable for the needs of real-time forecast. But availability of only dynamical characteristics is not enough for qualitative forecast of a thunderstorm. For this aim we need to obtain data about time evolution of cloud droplet vertical profile, i.e. to calculate space and time characteristics of droplet distribution function provided by microphysical block of the model.

Distribution function calculation demands new mesh generation which should define drop mass intervals or bins in each node of the dynamical space mesh. If dynamical mesh consists of the N nodes, appearance of microphysical block results in increasing of a total number of calculations at least by factor of $N \cdot N_1^2$, where N_1 is a number of bins. Taking into account that $N_1 \geq N$ the number of required calculations increases tremendously and so it takes already 30 seconds of computer time to calculate 60 minute cloud evolution cycle if $N_1 \approx N$, 90 seconds if $N_1 = 2N$ and 1100 seconds if $N_1 = 3N$. We should note that in order to obtain acceptable microphysical cloud characteristics one should take N_1 no less than 2·N. So we can see that addition of the microphysical block for calculation of only one distribution function increases total calculation time by factor of 3. If one adds distribution function not only for cloud droplets but for different forms of ice crystals, hail and graupel, total calculation time becomes quite unacceptable for real-time forecast. And the necessity of parallelization technique use becomes quite evident.

5 Parallelization Model

Numerical scheme for the dynamical part of the model is an explicit one. So we can easily calculate all dynamical characteristics of the cloud at a time step “n+1” if we know them in each node of the mesh at a time step “n”. And though to calculate dynamical characteristic in a mesh node “i” we should know corresponding characteristic in a neighbor mesh node “i-1”, or “i+1” we can easily do this as all necessary values have been already calculated in the previous time step. That is why we can use space parallelization [5-8] for our problem solution.

For this purpose we divide computational region of the model into several subsections (Fig.2)

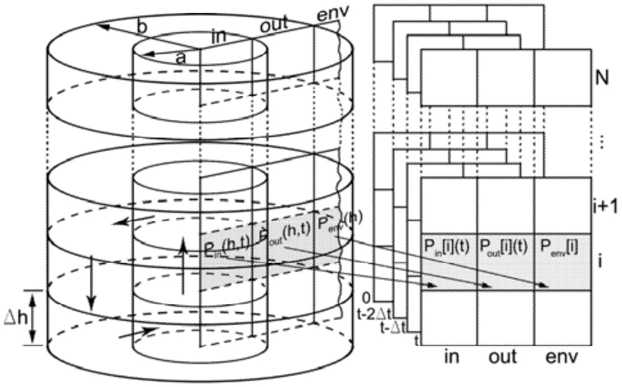


Fig. 2. Parallelization scheme of the model

Each subsection represents a cylinder of the height Δh and includes parts of the inner and outer cylinders as well as a part of the environment at rest.

Two methods of parallel calculations have been used. The first one supposes parallelization of only microphysical processes as the most time consuming. Dynamical part of the model is calculated within the entire computational region (big cylinder) while microphysical part is calculated in parallel within space subsections. The second method implies parallelization of both dynamical and microphysical model blocks, so all cloud characteristics are calculated within each space subsection.

Multi-thread technology was used to realize parallelization methodology. Threads are created, and the data calculated on the previous time step is passed to the threads. Each thread implements calculation within definite mesh nodes. The transfer to the next time step is implemented when all threads fulfill their calculations.

As each launch of the thread demands definite time, the number of threads should be diminished in order to decrease computational overheads. It is optimal to launch N threads for N core processor or 2N threads if the cores optimize 2 threads implementation, while using Hyper-Threading technology for example.

As at each time step processor should wait for completion of implementation of all threads, the problem of load balancing appears to be challenging. It is not easy to find the solution because calculation of cloud characteristics in different subsections demands quite different time due to the fact that it is not necessary to obtain microphysical characteristics in the mesh subsections where cloud droplets are absent and relative humidity is less than 100%. Special procedure of mesh subsection redistribution was used to obtain equal time of thread implementation. The procedure implies calculation in neighboring subsections in different threads and provides acceptable level of load balancing.

It should be noted that some parts of the model program, such as creation and launch of the thread, calculation of boundary characteristics are calculated in single-thread regime.

6 Calculation Results

The results of numerical simulation show that the model is capable to describe warm rain processes in convective clouds under various vertical distributions of temperature and relative humidity of the outer atmosphere. The model reproduces evolution of vertical velocity, mixing ration of cloud droplets and cloud droplet spectrum in time and space. It can predict maximum and minimum values of the above mentioned dynamical and microphysical characteristics and besides the values of the height of a cloud base and upper boundary, precipitation rate and total quantity of the rainfall. All that characteristics are of major value for prediction of dangerous convective cloud phenomena such as thunderstorms, hails and rain storms.

Besides numerical experiments targeted to obtain physical results essential attention has been paid for investigation of calculation effectiveness of the model and especially for investigation the effectiveness of parallelization.

Three types of processors were used for model calculations: K1(Core 2 Duo 6400, 2.13 GHz, 2.5 GB, 2 cores), K2 (Core 2 Quad Q8200, 2.33 GHz, 2.5 GB, 4 cores), K3 (Core 2 Quad Q6600, 2.4 GHz, 2.0 GB ,4 cores). Calculations were provided for different number of bins (drop mass intervals), different number of threads and the two methods of parallel calculations (parallelization of only microphysical processes and parallelization of both microphysical and dynamical processes). The results are presented in the tables 1-4.

Table 1. Calculation time (seconds) of 1 hr model cloud evolution obtained with the help of different types of processors (K1, K2, K3). Parallelization of only microphysical processes is considered. 4 threads are used. N^1 – is the number of bins.

N^1	50	70	100	150	250
K1	5,52	7,29	10,05	15,36	28,39
K2	4,19	4,94	6,16	8,78	15,02
K3	4,09	4,78	6,14	8,70	14,84

Table 2. Calculation time (seconds) of 1.5 min model cloud evolution obtained with the help of different number of threads (NTh) (processor K3). Parallelization of only microphysical processes is considered. N^1 – is the number of bins.

N^1	50	70	100
NTh = 1	15,60	36,19	89,14
NTh = 2	9,17	19,09	46,84
NTh = 3	13,05	17,09	34,80
NTh = 4	22,56	24,25	31,66

Table 3. Calculation time (seconds) of 1 hr model cloud evolution obtained with the help of different types of processors (K1, K2). Parallelization of both microphysical and dynamical processes is considered. 4 threads are used. N^1 – is the number of bins.

N^1	50	70	100	150	250
K1	5,14	6,73	9,44	14,30	22,62
K2	3,90	4,51	5,64	7,66	12,86

The results presented in the table 1 show that parallelization with the help of 4 core processors is more efficient than with the help of 2 core processor. Efficiency increases with increasing of the number of bins. Time difference between 2 and 4 core processors is about 20% in case of N^1=50 and is about 50% in case of N^1=250.

The results presented in the table 2 show that thread number influence is depended on the bin number (N^1). At the smallest value of N^1 the most effective is 2 threads using, at the biggest value 4 threads using is the most effective.

The results presented in the table 3 prove the above conclusion that 4 core processor is more efficient that 2 core. If we compare data in the tables 2 and 3 we can see that dynamical process parallelization contributes not so much in calculation time decrease (less than 10%). So the most time consuming part of the model is microphysical block which should be parallelized first of all.

And at last we present data (table 4) which characterize computational time of sequential algorithm. Comparison of the results in tables 4, 2 and 3 shows that speed up of cloud evolution calculation varies from 1,5 up to 3,0 dependent upon bin number.

Speed up is less than 4 (the number of threads) because of the time spent on thread creation and launch as well as on operations which should be provided in one thread regime.

Table 4. Calculation time (seconds) of 1 hr model cloud evolution obtained with the help of different types of processors (K1, K2). Without Parallelization. N^1 – is the number of bins.

N^1	50	70	100	150	250
K1	7,86	10,61	15,52	23,70	43,00
K2	6,27	8,80	12,47	19,38	37,36

7 Conclusions

1.5-D convective cloud model with detailed description of microphysical processes is presented in the paper. Possibilities of the model parallelization on multi-core processors with the different number of cores have been investigated. It is shown that parallelization with the help of 4 core processors is more efficient that with the help of 2 core processors. Multi-thread technology was used for realization of parallel algorithm. It is obtained that the number of threads should be equal or should be 2 times more than the number of processor cores. Comparison of the calculation results of sequential and parallel algorithms shows that speed up can vary from 1,5 to 3,0 in case of 4 parallel threads use. Investigation shows that use of rather complex numerical models for real-time forecast of dangerous convective phenomena is possible in case of realization of model parallelization on multi-core processors.

References

1. Asai, T., Kasahara, A.: A Theoretical Study of the Compensating Downward Motions Associated with Cumulus Clouds. Journal of the Atmospheric Sciences 24, 487–497 (1967)
2. Raba, N.O., Stankova, E.N.: Research of influence of compensating descending flow on cloud's life cycle by means of 1.5-dimensional model with 2 cylinders. In: Proceedings of MGO, vol. 559, pp. 192–209 (2009) (in Russian)
3. Khain, A., Pokrovsky, A., Pinsky, M.: Simulation of Effects of Atmospheric Aerosols on Deep Turbulent Convective Clouds Using a Spectral Microphysics Mixed-Phase Cumulus Cloud Model. Part I: Model Description and Possible Applications. Journal of the Atmospheric Sciences 61, 2963–2982 (2004)
4. Stankova, E.N., Zatevakhin, M.A.: The modified Kovetz and Olund method for the numerical solution of stochastic coalescence equation. In: Proceedings 12th International Conference on Clouds and Precipitation, Zurich, August 19-23, pp. 921–923 (1996)
5. Voevodin, V.V.: Informational structure of sequential programs. Russ. J. of Num. a An. and Math. Modelling 10(3), 279–286 (1995)
6. Voevodin, V.V.: Mathematical foundations of parallel computing. Series in computer science, vol. 33, p. 343. World Scientific Publishing Co, Singapore (1992)
7. Bogdanov, A.V., Korkhov, V.V., Mareev, V.V., Stankova, E.N.: Architectures and topologies of multiprocessor computational systems, p. 176 (2004) (in Russian)
8. Babb, R.G. (ed.): Programming Parallel Processors. Addison-Wesley Publishing Company, Reading (1988)

Transformation, Reduction and Extrapolation Techniques for Feynman Loop Integrals

Elise de Doncker[1], Junpei Fujimoto[2], Nobuyuki Hamaguchi[4], Tadashi Ishikawa[2], Yoshimasa Kurihara[2], Yoshimitsu Shimizu[3], and Fukuko Yuasa[2]

[1] Department of Computer Science,
Western Michigan University, U.S.A.
elise@cs.wmich.edu

[2] High Energy Accelerator Research Organization (KEK),
Oho 1-1, Tsukuba, Ibaraki, 305, Japan
{junpei.fujimoto,tadashi.ishikawa,yoshimasa.kurihara,
fukuko.yuasa}@kek.jp

[3] Graduate University for Advanced Studies, Hayama, Miura-gun,
Kanagawa, 240-0193, Japan
yoshimitsu.shimizu@kek.jp

[4] Hitachi, Ltd., Software Division, Totsuka-ku, Yokohama, Japan
nobuyuki.hamaguchi.sa@hitachi.com

Abstract. We address the computation of Feynman loop integrals, which are required for perturbation calculations in high energy physics, as they contribute corrections to the scattering amplitude for the collision of elementary particles. Results in this field can be used in the verification of theoretical models, compared with data measured at colliders.

We made a numerical computation feasible for various types of one and two-loop Feynman integrals, by parametrizing the integral to be computed and extrapolating to the limit as the parameter introduced in the denominator of the integrand tends to zero. In order to handle additional singularities at the boundaries of the integration domain, the extrapolation can be preceded by a transformation and/or by a sector decomposition. With the goal of demonstrating the applicability of the combined integration and extrapolation methods to a wide range of problems, we give a survey of earlier work and present additional applications with new results. We aim for an automatic or semi-automatic approach, in order to greatly reduce the amount of analytic manipulation required before the numeric approximation.

1 Introduction

Feynman diagrams represent configurations for particle interactions. Generally, with a given particle interaction, a large number of configurations may be associated, each corresponding with the probability of the configuration as a contribution to the cross section of the interaction. As such, a configuration corresponds with a contribution to a perturbation series expansion of the scattering amplitude, where zero-order contributions arise from tree configurations and higher order contributions from loop diagrams.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 139–154, 2010.

The latter require the computation of loop integrals, characterized by (threshold) singularities from vanishing integrand denominators in the interior of the integration region.

In earlier work we studied the application of *extrapolation* based techniques to the computation of such diagrams as one-loop 3-point (*vertex*), 4-point (*box*); and two-loop *ladder vertex* and *self-energy* diagrams. In this paper, we give results for one-loop *pentagon*, two-loop *crossed vertex* and two-loop *ladder box* diagrams.

The integrals are generally difficult to compute in view of the integrand singularities. We obtain the integral approximation in the limit as the value of a parameter (ε) in the integrand tends to zero. This involves the numerical evaluation of a sequence of integrals for decreasing ε, and a procedure for convergence acceleration or extrapolation to the limit.

A brief introduction to Feynman diagrams and loop integrals is given in Section 2. The numerical integration and extrapolation procedures which are relevant to this paper are outlined in Section 3, including a discussion of numerical adaptive and iterated integration. Section 4 derives a presentation of Feynman one-loop integrals and presents new results for the one-loop pentagon. The two-loop crossed and the ladder box integrals are derived in Section 5 and results are given.

In order to handle additional singularities at the boundaries of the integration domain, the extrapolation can be preceded by a transformation (as in the case of the two-loop vertex ladder (planar) diagram [13]) and/or by a sector decomposition, splitting the original problem and overlapping singularities over a sum of sector integrals (as for the two-loop crossed diagram), cf. Section 5.1. A reduction by sector decomposition can be coupled with extrapolation when infrared singularities are present (occurring as masses tend to zero), as well as threshold singularities in the interior of the domain.

An important aspect of the success of the singular integral evaluation itself is the numerical iterated integration method, applicable in some situations where standard multidimensional integration software fails. Various one- and multi-dimensional integration methods can be combined for different sets of coordinate directions. What we refer to as *iterated integration* is, in fact, implemented recursively and has as an added advantage that its memory use is not extensive in comparison with standard adaptive multi-dimensional integration software.

2 Feynman Diagrams and Loop Integrals

In high energy physics the computation of loop integrals is required to obtain higher order terms in perturbation calculations of the scattering amplitude. The latter in turn deliver corrections to the cross section for a collision of elementary particles, which corresponds to the probability of the given configuration in energy$-$momentum space (E, p^1, p^2, p^3). A Feynman diagram is a graph representation of a particle interaction. Each edge of the graph (*line* or *propagator*) is associated with an intermediate state of a particle, and particles meet at *vertices* according to a coupling constant g which indicates the strength of the interaction. An interaction generally corresponds to a large number of diagrams of different types. The term *loop* refers to the occurrence of one or multiple loops in the Feynman diagram.

The matrix element of one-loop corrections is assumed to be given by the real part of the product of a one-loop amplitude and the conjugate of a tree amplitude. Figure 1 (a)

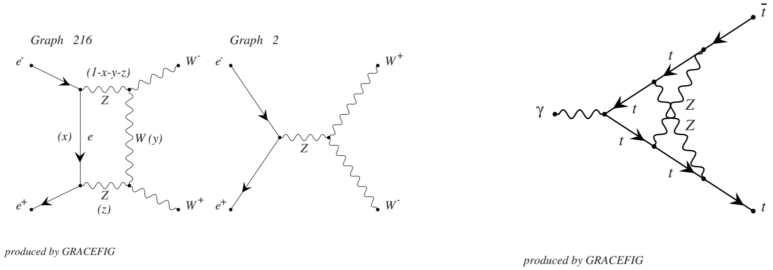


Fig. 1. (a) Box and tree diagrams for $e^- e^+ \to W^- \; W^+$; (b) Crossed vertex diagram

shows an example of a box diagram and a tree diagram of a Z boson exchange for the interaction $e^- e^+ \to W^- \; W^+$. The tree diagram shows the electron and positron colliding at the left vertex, while producing a Z boson which annihilates at the second vertex into the W^-, W^+ pair. Figure 1 (b) displays a sample two-loop (crossed vertex) diagram. The Feynman diagrams given in this paper and corresponding matrix elements for one-loop corrections are generated automatically by the GRACE-loop [2] system.

In a general form, loop integrals are given by

$$\mathcal{I}[\wp] = \int \prod_{\mu=1}^{L} \frac{d^4 l_\mu}{(2\pi)^4 i} \, \wp(k_1, \ldots, k_n) \prod_{\ell=1}^{n} \frac{1}{k_\ell^2 - m_\ell^2 + i\varepsilon}, \qquad (1)$$

where L is the number of loops, l_μ are the independent loop momenta and n is the number of propagators [21]. The momentum on the ℓ-th internal line is of the form $k_\ell = \sum_{\mu=1}^{L} a_{\ell\mu} l_\mu + P_\ell$, where the $a_{\ell\mu}$ are constants, P_ℓ is the sum of the external momenta flowing along the internal line ℓ, and the corresponding mass is $m_\ell,\ 1 \le \ell \le n$.

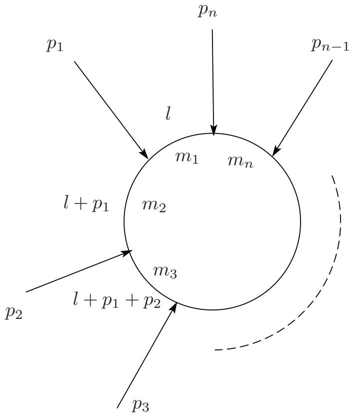


Fig. 2. One-loop diagram

Figure 2 shows a one-loop diagram with n external legs [20], where p_j is the incoming momentum of the j-th external particle, m_ℓ the mass carried by the ℓ-th internal line and the momentum is $l + \sum_{j=1}^{\ell-1} p_j$. A loop propagator gives rise to a factor in the integrand denominator which may cause an integrand singularity in the integration region. The term $i\varepsilon$ displaces the pole into the complex plane. The integral is then obtained by taking the limit as $\varepsilon \to 0$. A physical scattering amplitude contains this type of integrals and its value is defined at $\varepsilon = 0$. In (1), $\wp$ is generally a polynomial; $\mathcal{I}[1]$ is called a *scalar* integral.

For the simplest cases, results can be obtained analytically and expressed in terms of special functions. Numerically, the singularity can be avoided by a deformation of the integration contour away from the pole (cf., e.g., [32,31,1]). Numerical techniques have generally been successful only after considerable analytic manipulation (see also, e.g., [20,44,39,16,18]).

In [13] we presented a novel method for the computation of loop integrals based on numerical integration and extrapolation. The extrapolation is used to approximate the limit as $\varepsilon \to 0$ of a sequence of integrals computed for decreasing ε. We gave results for scalar one-loop vertex, one-loop box and two-loop planar vertex integrals. Apart from possibly a transformation preceding the extrapolation, the technique does not require analytic manipulation, nor knowledge of the location of the singularity. As expected, this *automatic* evaluation of loop integrals comes with a tradeoff with respect to speed. Even though in most cases the ε values needed for the sequence are not actually small, the computation time for the integrals increases as the singular behavior becomes more prevalent. As an important advantage, the method is also suitable for *tensor* (non-scalar) integrals without modification, as was shown for a non-scalar box diagram in [15]. Analytically, the non-scalar integral evaluation is generally reduced to that of a set of scalar integrals. Furthermore, the integration technique with extrapolation handles complex masses without added difficulty [49].

Loop integrals may show *infrared (IR)* singularities when masses are negligible. In that case the infinite part of the integral needs to be separated. This can be achieved by *dimensional regularization* and *sector decomposition* [28,5,4,1]. In this paper we use sector decomposition before the extrapolation for the crossed vertex diagram (where no IR divergence is present), with the goal of disentangling overlapping singularities. Sector decomposition and iterated integration followed by extrapolation is applied in [11,14,22].

We showed how to use the extrapolation method with photon regularization in [11] and with a modification of the technique in [47]. In photon regularization for a one-loop vertex diagram, the negligible mass M is replaced by a fictitious photon mass. We reported extrapolation results in [11] for $M = 10^{-5}$ GeV. It is noted that the procedure breaks down for $M = 10^{-9}$ GeV in double precision (in view of the fact that the ε parameter in the denominator dominates the behavior for small M). By using quadruple precision, the point of deterioration can be moved. For example, results of up to 8-digit accuracy are obtained using quadruple precision for the one-loop box diagram with $M = 10^{-15}$ GeV. Using the extended precision library HMLIB [27] (based on the IEEE 754-1985 FP standard), results are given in [48,47] for the one-loop vertex diagram and M as small as $M = 10^{-160}$, and for the box diagram with $M \geq 10^{-30}$.

3 Numerical Techniques

3.1 Adaptive Iterated Integration

In numerical integration it is our goal to obtain an approximation Qf to a multivariate integral, $\mathcal{I}f = \int_{\mathcal{D}} f(\mathbf{x})\, d\mathbf{x}$ and an (absolute) error estimate Ef, intended to bound the achieved accuracy and not exceeding the tolerated error. The integration domain $\mathcal{D}$ is usually a *standard* region (to which the given integral may have been transformed previously), such as the unit cube or simplex.

Assume the integral $\mathcal{I}f$ of the N-variate function $f(\mathbf{x})$ can be written as

$$\mathcal{I}f = \int_{\mathcal{D}_1} d\mathbf{x}^{(1)} \dots \int_{\mathcal{D}_\ell} dx^{(\ell)} f(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(\ell)}) \tag{2}$$

such that $\mathcal{D} = \mathcal{D}_1 \times \dots \times \mathcal{D}_\ell$ is of dimension N; then we call $\mathcal{I}f$ an *iterated integral* for $\ell \geq 2$. The terminology is somewhat misleading as, in fact, the implementation will be of a recursive form. As an example, the integral $\mathcal{I}f = \int_a^b dx \int_\alpha^\beta dy \int_A^B dz\ f(x, y, z)$ over the 3D hyperrectangle $[a, b] \times [\alpha, \beta] \times [A, B]$ is represented as a 1D×2D integral in the form $\mathcal{I}g = \int_a^b \mathrm{d}x\ g(x)$, where $g(x) = \int_\alpha^\beta dy\ h(x, y)$.

The integrals over the regions $\mathcal{D}_j$, $1 \leq j \leq \ell$ in (2) can often be evaluated adaptively by available software. Adaptive partitioning procedures refine the function domain by selecting highest error regions for subsequent subdivision, thus concentrating function evaluations in difficult regions. Figure 3 shows the subdivision pattern in the (3D) domain of a box integral in the course of the integration, while "discovering" the surface where the integrand is singular [34].

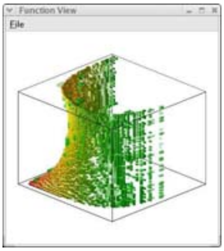


Fig. 3. Singularity surface in the 3D domain of a box integral

The local integral estimate on each subregion is calculated using a quadrature or cubature rule, or a pair or sequence of such rules. For example, a pair consisting of a Gauss rule and an interlacing Kronrod rule is used as a basis rule pair in the one-dimensional adaptive quadrature routine DQAGE in QUADPACK [40]. The local error is then estimated as a function of the difference between the rule values. Similarly, the multi-dimensional integration code DCUHRE [3] implements a family of cubature rules for the subregion integrations.

A priority queue such as a heap or linked list is maintained on the subregions, keyed with respect to the size of their local error. At each iteration, the worst subregion is selected for further subdivision and for integration over its newly formed subregions. Region partitioning continues until either the requested accuracy or a user-specified limit on the allowed work is reached. We measure the effort spent in the integration by means of the total number of integrand evaluations performed in the course of the procedure.

The work space needed for storing the subregion information in a standard adaptive multidimensional integration code can be very large when extensive partitioning is needed. For example, with the degree 7 cubature rule for 2D in DCUHRE, the number of function evaluations per region is $\mathrm{num} = 1+6N+2N(N-1)+2^N = 21$ for dimension $N = 2$. For a maximum allowed number of function evaluations of $\mathrm{maxpts} = 100$ million, the maximum number of regions that can be generated by successive bisections is $\mathrm{maxreg} = (\mathrm{maxpts} - \mathrm{num})/(2\ \mathrm{num}) + 1 = 2,380,952$ and the work space (supplied in the call to DCUHRE via a work array in Fortran) needs to be at least 19,047,634 doubles.

In comparison, the amount of space needed for region storage in a 1D×1D "recursive" integration with DQAGE×DQAGE is determined by the number of evaluations allowed in each coordinate direction, which is 10^4 evaluations for a maximum of 10^8 in two dimensions. Using the 15-point Gauss-Kronrod rule pair, the maximum number of intervals in each dimension is maxreg $= 333$, requiring an array space of about 1,500 doubles on each integration level (or a total of about 3,000).

3.2 Numerical Extrapolation

Methods for extrapolation to the limit $\mathcal{S}$ of a sequence $S(\varepsilon)$, as the parameter ε tends to 0, rely on the existence of an asymptotic expansion

$$S(\varepsilon) \sim \mathcal{S} + a_1\varphi_1(\varepsilon) + a_2\varphi_2(\varepsilon) + \ldots$$

Given a sequence $\{S(\varepsilon_\ell)\}$, an extrapolation is performed with the goal of creating sequences that converge to the limit $\mathcal{S}$ faster than the original sequence. Extrapolation methods and their application to numerical integration have been of considerable interest in the literature (see, e.g., [24,35,36,37,10,40,42,26,6,33,19]).

$$\begin{array}{ccccc}
 & \tau_{00} & & & \\
0 & & \tau_{01} & & \\
 & \tau_{10} & & \tau_{02} & \\
0 & & \tau_{11} & & \cdots \\
 & & \cdots & & \cdots \\
 & & \cdots & & \cdots \\
0 & & \tau_{\kappa-1,1} & & \cdots \\
 & \tau_{\kappa 0} & & \tau_{\kappa-1,2} & \\
0 & & \tau_{\kappa 1} & & \\
 & \tau_{\kappa+1,0} & & &
\end{array}$$

Fig. 4. ε-algorithm table

A *linear extrapolation* method solves (implicitly or explicitly) linear systems of the form

$$S(\varepsilon_\ell) = c_0 + c_1\varphi_1(\varepsilon_\ell) + \ldots + c_\nu\varphi_\nu(\varepsilon_\ell), \qquad \ell = 0, \ldots, \nu; \tag{3}$$

i.e., systems of order $(\nu + 1) \times (\nu + 1)$ in the unknowns $c_0, \ldots, c_\nu$ are solved for increasing values of ν. Note that the coefficients $\varphi_k(\varepsilon_\ell)$ need to be known explicitly in order to apply this method. As an example, the computation for $\varphi_k(\varepsilon) = \varepsilon^k$ can be carried out recursively using Richardson extrapolation [9].

For *non-linear* extrapolation we have made ample use of the ε-algorithm [41,46], which implements a sequence-to-sequence transformation recursively. A triangular table is computed as shown in Figure 4, according to the following recurrence:

$$\begin{aligned}
\tau_{\kappa,-1} &= 0 \\
\tau_{\kappa 0} &= \beta_\kappa \\
\tau_{\kappa,\lambda+1} &= \tau_{\kappa+1,\lambda-1} + \frac{1}{\tau_{\kappa+1,\lambda} - \tau_{\kappa\lambda}},
\end{aligned}$$

for a given entry (original) sequence β_κ, $\kappa = 0, 1, \ldots$. The even numbered columns $(\lambda = 0, 2, 4 \ldots)$ form transformed sequences, which are expected to converge faster than the original sequence under certain conditions. The ε-algorithm can be applied under more general conditions than those for linear extrapolation, for example, if the φ functions of (3) are of the form $\varphi_k(\varepsilon) = \varepsilon^{\alpha_k} \log^{\iota_k}(\varepsilon)$, where $\iota_k \geq 0$ integer and $\alpha_k > 0$ real and if a geometric sequence is used for ε.

4 One-Loop Integrals

4.1 Representation

The integral associated with the one-loop diagram of Figure 2 is represented by

$$\mathcal{I}_n = \int \frac{d^4 l}{(2\pi)^4 i} \frac{1}{\prod_{\ell=1}^{n}((l + \sum_{j=1}^{\ell-1} p_j)^2 - m_\ell^2 + i\varepsilon)}. \tag{4}$$

Using the identity

$$\frac{1}{\prod_{\ell=1}^{n} A_\ell} = \Gamma(n) \int_{\mathcal{S}_n} d\mathbf{x} \frac{\delta(1 - \sum_{\ell=1}^{n} x_\ell)}{(\sum_{\ell=1}^{n} x_\ell A_\ell)^n}. \tag{5}$$

and $\sum_{\ell=1}^{n} x_\ell = 1$ (to express one of the variables x_ℓ in terms of the $n-1$ other ones) yields (4) in the form

$$\mathcal{I}_n = \Gamma(n) \int_{\mathcal{S}_{n-1}} d\mathbf{x} \int \frac{d^4 l}{(2\pi)^4 i} \frac{1}{(l^2 - \mathcal{D}_n(\mathbf{x}) + i\varepsilon)^n}$$

where $\mathcal{D}_n(\mathbf{x}) = \mathbf{x}^\tau B\mathbf{x} + 2\mathbf{v} \cdot \mathbf{x} + C$ is a quadratic with coefficients determined by the momenta and masses. Integrating over l gives

$$\Gamma(n) \int \frac{d^4 l}{(2\pi)^4 i} \frac{1}{(l^2 - \mathcal{D}_n(\mathbf{x}) + i\varepsilon)^n} = \frac{1}{(4\pi)^2} \frac{(-1)^n}{(\mathcal{D}_n(\mathbf{x}) - i\varepsilon)^{n-2}} \Gamma(n-2),$$

resulting in

$$\mathcal{I}_n = \frac{(-1)^n \Gamma(n-2)}{(4\pi)^2} \int_{\mathcal{S}_{n-1}} d\mathbf{x} \frac{1}{(\mathcal{D}_n(\mathbf{x}) - i\varepsilon)^{n-2}}. \tag{6}$$

Figure 5 displays examples of a one-loop (a) vertex, (b) box diagram for $e^- e^+ \to t\bar{t}$ and (c) pentagon diagram for $e^- e^+ \to e^- e^+ Z$. For the vertex diagram of Figure 5 (a), integral (6) becomes

$$\mathcal{I}_3 = -\frac{1}{16\pi^2} \int_0^1 dx_1 \int_0^{1-x_1} dx_2 \frac{1}{\mathcal{D}_3(x_1, x_2) - i\varepsilon},$$

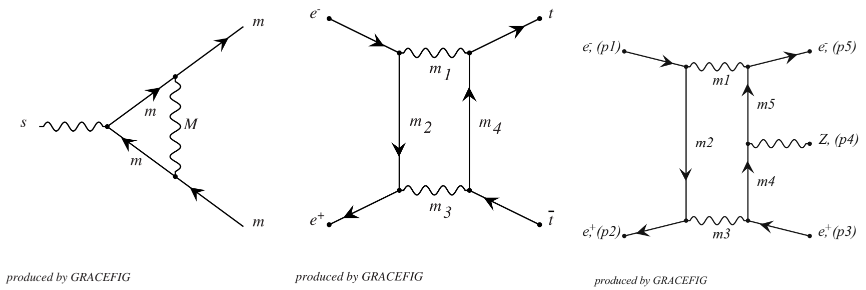
where

$$\mathcal{D}_3(x_1, x_2) = -s x_1 x_2 + m^2 (x_1 + x_2)^2 + M^2 (1 - x_1 - x_2)$$

and s denotes the squared energy. The integral for the box (4-point) diagram of Figure 5 (b) is given by (6) with

$$\mathcal{D}_4(\mathbf{x}) = \mathbf{x}^\tau B\mathbf{x} + 2\mathbf{v} \cdot \mathbf{x} + C,$$

$B_{\iota j} = q_\iota q_j,\ q_1 = -p_1,\ q_2 = p_2,\ q_3 = p_2 + p_3,\ C = M_0^2 = m_2^2$ and $v_\iota = \frac{1}{2}(-q_\iota^2 + M_\iota^2 - M_0^2)$ with $M_1 = m_1,\ M_2 = m_3,\ M_3 = m_4$. We gave results for several cases in previous work, e.g., [13,15]. The computations take time of the order of seconds for the vertex problem and around 20 minutes for the box problem, on a laptop with a single 2.5 GHz (Pentium) processor. An accuracy of 15 digits was reached for the vertex problem in double precision.



(a) (b) (c)

Fig. 5. (a) Generic one-loop vertex; (b) Box diagram; (c) Pentagon diagram

4.2 One-Loop Pentagon Diagram

For the one-loop pentagon diagram, using $x_2 = 1 - x_1 - x_3 - x_4 - x_5$ in (5) and introducing $l' = l - x_1 p_1 + x_3 p_2 + x_4 p_3 + x_5 p_4$ in the expression of $\sum_{\ell=1}^{n} x_\ell d_\ell$ in the denominator, allows re-grouping the terms into the form $(l'^2 - \mathcal{D}_5(\mathbf{x}))$ where $\mathcal{D}_5(\mathbf{x})$ is written as a quadratic form,

$$\mathcal{D}_5(\mathbf{x}) = \mathbf{x}^\tau B\mathbf{x} + 2\mathbf{v} \cdot \mathbf{x} + C$$

with $B_{\iota j} = q_\iota q_j,\ q_1 = -p_1,\ q_2 = p_2,\ q_3 = p_2 + p_3,\ q_4 = p_2 + p_3 + p_4,\ C = M_0^2 = m_2^2$ and $v_\iota = \frac{1}{2}(-q_\iota^2 + M_\iota^2 - M_0^2)$; $M_1 = m_1$ and $M_j = m_{j+1}$ for $j > 1$.

Integration over the loop momentum then yields $\mathcal{I}_5$ as given in (6) with

$$\begin{aligned}\mathcal{D}_5(x_1, x_3, x_4, x_5) = {} & M_Z^2(x_1 + x_3 - x_4 x_5) \\ & + m_e^2(x_1^2 + 2x_1 x_3 + x_1 x_4 - 2x_1 + x_3^2 + x_3 x_5 - 2x_3 + 1) \\ & - x_1 x_3 s_{12} - x_3 x_5 s_{34} - x_1 x_4 s_{45} \\ & - (1 - x_1 - x_3 - x_4 - x_5)(x_4 s_{23+} x_5 s_{51}),\end{aligned}$$

where $s_{\iota j} = p_\iota p_j = B_{\iota j}$, $m_1 = m_3 = M_Z$ and $m_2 = m_4 = m_5 = m_e$. The table of Figure 6 shows an extrapolation for the one-loop pentagon diagram of Figure 5 (c) with parameters $M_Z = 90$ GeV, $m_e = 0.5 \times 10^{-3}$ GeV, $s_{12} = 100000$ GeV^2, $s_{23} = -30471$ GeV^2, $s_{34} = 32384$ GeV^2, $s_{45} = 37834$ GeV^2, $s_{51} = -14146$ GeV^2 [49], using the 1D adaptive integration program DQAGE [40] with the 15-point Gauss-Kronrod basis rule pair. The result agrees with the data for the real value obtained in [23] with a sector decomposition [28,5,4], using the symbolic manipulation system FORM [45]. The value listed in that paper is 0.411918E-13 with an absolute error estimate of 0.269E-17. The ε-values for the integral approximations $Q(\varepsilon)f$ in the second column of Figure 6 are computed for the range of $\varepsilon_j = 1.2^{-j}$, $j = 30, \ldots, 24$. The integrals are approximated to a relative error tolerance of 10^{-5}. Note that the entry sequence in column 2 is still far from the value for the limit obtained on the right of the table.

j	$Q(\varepsilon_j)$			
30	0.49428E-14			
29	0.10453E-13	0.48434E-13		
28	0.15264E-13	0.45061E-13	0.40935E-13	
27	0.19407E-13	0.43314E-13	0.41077E-13	**0.41188**E-13
26	0.22938E-13	0.42371E-13	**0.411**38E-13	
25	0.25926E-13	**0.41**851E-13		
24	0.28442E-13			

Fig. 6. One-loop pentagon extrapolation table

We listed the table showing seven extrapolations. The result did not improve by much after that in view of the fact that the $Q(\varepsilon)f$ were computed to about 5-digit accuracy. The time is of the order of 10,000 seconds on a 3.0 GHz system. We also performed the computation by replacing $x_5 = 1 - x_1 - x_2 - x_3 - x_4$ in (5); it gave similar results but took some more time.

5 Two-Loop Diagrams

5.1 Two-Loop Crossed (Non-planar) Vertex

We presented results in [12] for the scalar two-loop crossed vertex process. It has important applications, e.g., for the study of the b-s-γ process [8,29,38,7].

Derivation. The integral as given in [32] is

$$\mathcal{I} = \frac{1}{8}\int_0^1 dz_1\, dz_2\, dz_3\, \delta(1 - \Sigma_{j=1}^3 z_j)\; z_1 z_2 z_3 \int_{-1}^1 dy_1\, dy_2\, dy_3\, \frac{1}{(D_3 - i\varepsilon)^2},$$

where D_3 is a quadratic in $\mathbf{y} = (y_1, y_2, y_3)^\tau$, and depends on the masses m_j, $1 \le j \le 6$ and on $s_\ell = p_\ell^2$, $\ell = 1, 2, 3$. Specifically, $D_3 = \boldsymbol{y}^\tau A \boldsymbol{y} + \boldsymbol{b}^\tau \boldsymbol{y} + c$, where

$$A = \frac{1}{4}\begin{pmatrix} -z_1^2(z_2+z_3)s_1 & z_1z_2z_3(-s_1-s_2+s_3)/2 & z_1z_2z_3(-s_1+s_2-s_3)/2 \\ z_1z_2z_3(-s_1-s_2+s_3)/2 & -z_2^2(z_3+z_1)s_2 & z_1z_2z_3(s_1-s_2-s_3)/2 \\ z_1z_2z_3(-s_1+s_2-s_3)/2 & z_1z_2z_3(s_1-s_2-s_3)/2 & -z_3^2(z_1+z_2)s_3 \end{pmatrix},$$

$$\boldsymbol{b} = \frac{1}{2}\,U\begin{pmatrix} z_1(m_3^2 - m_4^2) \\ z_2(m_5^2 - m_6^2) \\ z_3(m_2^2 - m_1^2) \end{pmatrix},$$

$$c = \frac{1}{4}\,U(z_1s_1 + z_2s_2 + z_3s_3 - 2(m_3^2+m_4^2)z_1 - 2(m_5^2+m_6^2)z_2 - 2(m_1^2+m_2^2)z_3)$$

and $U = z_1z_2 + z_2z_3 + z_3z_1$.

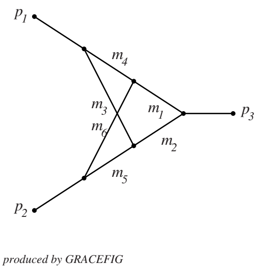


Fig. 7. Two-loop crossed vertex

We apply a transformation/reduction which was used to handle infrared divergent loop integrals in [4]. This is a sector decomposition [28,5], which casts the integral $\mathcal{I}f$ in the form $\mathcal{I}f = \mathcal{I}^{(1)}F + \mathcal{I}^{(2)}F + \mathcal{I}^{(3)}F$, where $F = F(\boldsymbol{z})$ represents the inner integral and

$$\mathcal{I}^{(1)}F = \int_0^1 dz_1 \int_0^{z_1} dz_2 \int_0^{z_1} dz_3 \; F(\boldsymbol{z}),$$

$$\mathcal{I}^{(2)}F = \int_0^1 dz_2 \int_0^{z_2} dz_1 \int_0^{z_2} dz_3 \; F(\boldsymbol{z}),$$

$$\mathcal{I}^{(3)}F = \int_0^1 dz_3 \int_0^{z_3} dz_1 \int_0^{z_3} dz_2 \; F(\boldsymbol{z}).$$

$\mathcal{I}^{(1)}F$ is transformed according to $z_2 = t_1 z_1, \;\; z_3 = t_2 z_1$, which yields

$$\mathcal{I}^{(1)}F = \frac{1}{8}\int_0^1 dz_1 \int_0^1 dt_1 \int_0^1 dt_2 \, t_1 t_2 \;\, \delta(1 - z_1(1 + t_1 + t_2)) \, z_1^5 \int_{-1}^1 d\mathbf{y} \, \frac{1}{(D_3 - i\varepsilon)^2}$$

where $D_3 = z_1^3(A_1 + B_1 + C_1)$ and

$$A_1 = \frac{1}{4}\,\mathbf{y}^\tau \begin{pmatrix} -(t_1+t_2)s_1 & t_1t_2(-s1-s2+s3)/2 & t_1t_2(-s_1+s_2-s_3)/2 \\ t_1t_2(-s_1-s_2+s_3)/2 & -\,t_1^2(t_2+1)s_2 & t_1t_2(s_1-s_2-s_3)/2 \\ t_1t_2(-s_1+s_2-s_3)/2 & t_1t_2(s_1-s_2-s_3)/2 & -\,t_2^2(1+t_1)s_3 \end{pmatrix} \mathbf{y},$$

$$B_1 = \frac{1}{2}(t_1 + t_1t_2 + t_2) \begin{pmatrix} m_3^2 - m_4^2 \\ (m_5^2 - m_6^2)t_1 \\ (m_2^2 - m_1^2)t_2 \end{pmatrix}^\tau \mathbf{y},$$

$$C_1 = \frac{1}{4}(t_1{+}t_1t_2{+}t_2)(s_1{+}t_1s_2{+}t_2s_3{-}2(m_3^2{-}m_4^2){-}2(m_5^2{-}m_6^2)t_1{-}2(m_2^2{-}m_1^2)t_2).$$

After splitting $z_1^5/(D_3 - i\varepsilon)^2$ into its real and imaginary part, we transform $z_1 = u_1/(1 + t_1 + t_2)$, so that $dz_1/z_1 = du_1/u_1$ and $\delta(1 - z_1(1 + t_1 + t_2)) = \delta(1 - u_1)$; and the integration in u_1 reduces to setting $u_1 = 1$ in the integrand. We find

$$\begin{aligned}\mathcal{I}^{(1)}F = \frac{1}{8}\int_0^1 dt_1 \int_0^1 dt_2 \int_{-1}^1 d\mathbf{y} \;\Bigg[&\frac{(A_1 + B_1 + C_1)^2 - \varepsilon^2(1 + t_1 + t_2)^6}{((A_1 + B_1 + C_1)^2 + \varepsilon^2(1 + t_1 + t_2)^6)^2} \Bigg] \\ &+ \frac{2i\varepsilon(A_1 + B_1 + C_1)(1 + t_1 + t_2)^3}{((A_1 + B_1 + C_1)^2 + \varepsilon^2(1 + t_1 + t_2)^6)^2}. \end{aligned} \tag{7}$$

$\mathcal{I}^{(2)}F$ and $\mathcal{I}^{(3)}F$ are derived in a similar manner.

Numerical Integration and Extrapolation Results. We apply iterated adaptive integration together with extrapolation [13,15] to compute the integral. The 5D integral is treated as a 2D×1D×1D×1D problem. The inner three dimensions need substantial partitioning in view of the quadratic hypersurface singularity. As an alternative approach, the original problem can be treated as a $(1D)^5$-iterated integral.

j	$Q(\varepsilon_j)$			
32	0.1019E-08			
31	0.1096E-08	0.1480E-08		
30	0.1160E-08	0.1411E-08	0.1441E-08	
29	0.1211E-08	0.1478E-08	0.1469E-08	**0.1464**E-08
28	0.1254E-08	0.1468E-08	**0.14**51E-08	
27	0.1290E-08	**0.146**2E-08		
26	0.1319E-08			

Fig. 8. Crossed vertex extrapolation table

We use the 1D QUADPACK [40] adaptive routine DQAGE, with the 15-point Gauss-Kronrod basis quadrature rule pair. The multivariate integration is based on DCUHRE [25,3] and its cubature rule of polynomial degree 7 for integration over the subregions.

The table of Figure 8 shows an extrapolation obtained for the two-loop crossed vertex problem of Figure 7, with parameters $m_1 = m_2 = m_4 = m_5 = 150$ GeV, $m_3 = m_6 = 91.17$ GeV; $s_1 = s_2 = 150^2$ GeV^2 and $s_3/m_1^2 = 5$. The $Q(\varepsilon_j)$ are numerical approximations to the 2D×1D×1D×1D integral for a requested relative tolerance of 10^{-3}. The extrapolation is performed with $\varepsilon_j = \epsilon^j$ where $\epsilon = 1.2$ and $j = 32, \ldots, 26$. The result agrees with the data in [32].

The tables of Figure 9 give new results obtained with the $(1D)^5$-iterated approach for parameters $m_1 = m_2 = m_3 = m_4 = m_5 = m_6 = m = 150$ GeV; $s_1 = s_2 = 0$ (real part).

5.2 Two-Loop Ladder (Planar) Box

In this section we address the calculation of the two-loop ladder box. The corresponding diagram is given in Figure 10 (a). The loop integral is given by

$$\mathcal{I} = \int_0^1 dx_1\, dx_2\, dx_3\, dx_4\, dx_5\, dx_6\, dx_7\, \delta(1 - \sum_{\ell=1}^{7} x_\ell) \frac{\mathcal{C}}{(\mathcal{D} + i\epsilon\mathcal{C})^3}.$$

With $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 1$ and the transformation $(x_1, x_2, x_3, x_4, x_5, x_6) \to (\rho, \xi, y_1, y_2, y_3, y_4)$ defined as

$$x_1 = \rho_1 u_1,\ \ x_2 = \rho_1 u_2,\ \ x_3 = \rho_1(1 - u_1 - u_2),\ \ x_4 = 1 - \rho_1 - \rho_2,\ \ x_5 = \rho_2 u_5,\ \ x_6 = \rho_2 u_6$$

and $\rho_1 = \rho\xi,\ \ \rho_2 = \rho(1 - \xi)$, with Jacobian $\rho^5 \xi^2 (1 - \xi)^2$, the integral is

$$\mathcal{I} = \int_0^1 d\rho\, d\xi \int_0^1 dy_1\, dy_2\, dy_3\, dy_4\, \frac{\mathcal{C}}{(\mathcal{D}_4 + i\epsilon\mathcal{C})^3}\, \rho^3 \xi^2 (1 - \xi)^2\, (1 - y_1)(1 - y_3), \tag{8}$$

where $\mathcal{D}_4$ is a quadratic in $\mathbf{y} = (y_1, y_2, y_3, y_4)^\tau$,

$$\mathcal{D}_4 = \mathbf{y}^\tau A \mathbf{y} + \mathbf{b}^\tau \mathbf{y} + c,$$

	Tarasov [43]	Ferroglia [17]	KEK
s_3/m^2	(hep ph/9505277)	(hep ph/0311186)	Minami Tateya
4.0	0.733120(0.02)	0.7331(1)	0.733120(2)
4.5	0.61644824(0.1)	0.6216(78)	0.61650(2)
5.0	0.5184444(0.3)	0.5203(40)	0.51845(1)
8.0	0.14555(0.7)	0.1455(20)	0.1455223(5)
20.0	-0.2047(0.8)	-0.2058(5)	-0.20471(4)
100.0	-0.0382(3)	-0.0385(1)	-0.0382(2)

(a)

	Tarasov [43]	Ferroglia [17]	KEK
s_3/m^2	(hep ph/9505277)	(hep ph/0311186)	Minami Tateya
4.5	-0.3349475(1)	-0.3402(71)	-0.3349(1)
5.0	-0.430997(0.3)	-0.4442(93)	-0.43100(5)
8.0	-0.5460(0.5)	-0.5491(40)	-0.54594(1)
20.0	-0.1876(4)	-0.1864(4)	-0.187578(10)

(b)

Fig. 9. (a) Real part; (b) Imaginary part (in units of 10^{-9})

and $\mathcal{C}(\rho, \xi) = -\rho\xi^2 + \rho\xi - \rho + 1$.

A is a cubic in ρ and ξ and depends on the masses m_j, $1 \leq j \leq 7$ and on the kinematical variables, $s = (p_1 + p_2)^2 = (p_3 + p_4)^2$ and $t = (p_1 - p_3)^2 = (p_2 - p_4)^2$. For $m_1 = m_2 = m_5 = m_6 = m$ and $m_3 = m_4 = m_7 = M$, we have

$$A =$$

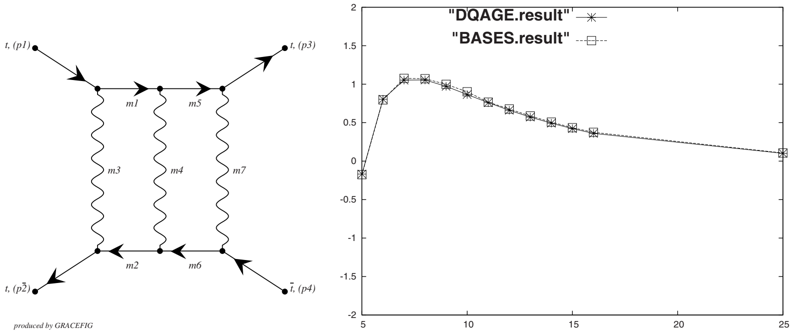
$$\begin{pmatrix} -m^2\rho\xi^2 w_0 & (s - 2m^2)\rho\xi^2 w_0 & (t - 2m^2)\rho\xi w_1 & (s + t - 2m^2)\rho\xi w_1 \\ (s - 2m^2)\rho\xi^2 w_0 & -m^2\rho\xi^2 w_0 & (s + t - 2m^2)\rho\xi w_1 & (t - 2m^2)\rho\xi w_1 \\ (t - 2m^2)\rho\xi w_1 & (s + t - 2m^2)\rho\xi w_1 & -m^2\rho(1 - \xi)^2 w_2 & (s - 2m^2)\rho(1 - \xi)^2 w_2 \\ (s + t - 2m^2)\rho\xi w_1 & (t - 2m^2)\rho\xi w_1 & (s - 2m^2)\rho(1 - \xi)^2 w_2 & -m^2\rho(1 - \xi)^2 w_2 \end{pmatrix},$$

with $w_0 = 1 - \rho\xi, \quad w_1 = (1 - \rho)(1 - \xi), \quad w_2 = (1 - \rho(1 - \xi))$;

$$\boldsymbol{b} = \begin{pmatrix} \rho\xi(-tw_1 + M^2 w_3) \\ \rho\xi(-tw_1 + M^2 w_3) \\ \rho(-t\xi w_1 + M^2 w_4) \\ \rho(-t\xi w_1 + M^2 w_4) \end{pmatrix},$$

with $w_3 = \xi w_0 + w_1$, $w_4 = (1 - \xi)(-\rho((1 - \xi)^2 + \xi) + 1)$; and $c = t\rho\xi w_1 - M^2 w_3$.

Figure 10 (b) shows numerical results obtained for the two-loop ladder box with parameters $m = 50$ GeV and $M = 90$ GeV; $t = -100^2$ GeV^2 and for s such that $5 \leq f_s = \frac{s}{m^2} \leq 25$. The real part integrals are plotted as a function of f_s. Indicated by asterisks ($*$) are the results obtained with a $(1D)^6$-iterated integration by the adaptive QUADPACK routine DQAGE with its 7- and 15-point Gauss-Kronrod quadrature rule pair. The results agree with independent calculations where the integrand is reduced to



(a) (b)

Fig. 10. (a) Two-loop planar box (b) Real part integral vs. $f_s = \frac{s}{m^2}$

the logarithmic level, followed by an adaptive Monte Carlo integration with the program BASES [30].

For a quadratic form, $D = \mathbf{x}^\tau A\mathbf{x} + \mathbf{b}^\tau \mathbf{x} + C$ of order N, this reduction (see, e.g., [44]) is based on differentiation properties of the form,

$$\begin{aligned} \frac{\Delta_N}{D^{n+1}} &= \frac{-4 + 2N/n}{D^n} - \frac{1}{n}\,\nabla^\tau\,(\frac{\mathbf{X}}{D^n}), \ \ \text{if } n > 0, \\ &= (-4 - 2N\log D) + \nabla^\tau(\mathbf{X}\log D), \ \ \text{if } n = 0 \end{aligned}$$

where $\mathbf{X} = 2\mathbf{x} + A^{-1}\mathbf{b} = A^{-1}\nabla D(\mathbf{x})$, $\nabla^\tau = (\partial_1, \partial_2, \ldots, \partial_N)$ and $\Delta_N = \mathbf{b}^\tau A^{-1}\mathbf{b} - 4C$ is the discriminant of D. It is assumed that $\Delta_N \neq 0$ and the matrix A is invertible. As a result of these formulas, the power of the quadratic in the denominator of the integrand of the loop integral can be reduced and $1/D$ can eventually be replaced in terms of $\log D$, in order to reduce the severity of the integrand singularity. After reduction, the BASES Monte Carlo procedure converges for the ladder box problem within an acceptable accuracy using 10^8 sample points. The BASES values are marked with squares (□) in Figure 10 (b).

We conclude that the *multivariate integration with extrapolation* technique enabled us to handle the ladder box problem at hand in an automatic way, for mass parameters and external momenta throughout the physical region.

6 Conclusions

In this paper we address the computation of one and two-loop integrals in high-energy physics using numerical iterated integration and extrapolation methods. Iterated adaptive integration is suitable in low dimensions and often outperforms standard numerical multivariate integration methods by eliminating or reducing the severity of integrand problems through the inner integration. The memory requirements which may become an issue with adaptive partitioning methods are reduced significantly.

We apply numerical extrapolation or convergence acceleration to approximate loop integrals in the limit as a threshold parameter in the interior of the integration region tends to zero. For several classes of loop integrals, this allows dealing with the corresponding threshold singularity without precise information on the nature and the location of the problem. For cases where singularities are present at the boundary of the integration region, we can apply transformations to smoothen the integrand behavior (as in the case of the two-loop ladder vertex and ladder box diagrams). Overlapping singularities can also be split over sectors of the integration region, which we used for the crossed vertex diagram. Sector decomposition and dimensional regularization can be used to separate the finite part integral when infrared divergence is present, and followed by an extrapolation to address the threshold singularity.

We presented results of combined transformation, reduction and extrapolation techniques for two-loop crossed vertex and ladder box diagrams. Whereas with an increased computational expense, the methods deliver an automatic or semi-automatic computation procedure. In future work we will look further into utilizing symbolic manipulation to a certain level, while still handling a large part of the problem numerically.

Acknowledgments. This work is partially supported by Grant-in-Aid for Scientific Research(B) (20340063) and Grant-in-Aid for Scientific Research on Innovative Areas (21105513). The authors thank the reviewers of this paper for their valuable comments.

References

1. Anastasiou, C., Beerli, S., Daleo, A.: Evaluating multi-loop Feynman diagrams with infrared and threshold singularities numerically. JHEP 0705, 71 (2005)
2. Bélanger, G., Boudjema, F., Fujimoto, J., Ishikawa, T., Kaneko, T., Kato, K., Shimizu, Y.: Automatic calculations in high energy physics and GRACE at one-loop. Physics Reports 430, 117–209 (2006)
3. Berntsen, J., Espelid, T.O., Genz, A.: Algorithm 698: DCUHRE-an adaptive multidimensional integration routine for a vector of integrals. ACM Trans. Math. Softw. 17, 452–456 (1991)
4. Binoth, T., Heinrich, G.: An automized algorithm to compute infrared divergent multi-loop integrals. Nuclear Physics B 585, 741–759 (2000)
5. Bollini, C.G., Giambiagi, J.J.: Dimensional renormalization: the number of dimensions as a regularizing parameter. Nuovo Cimento B 12 20 (1972)
6. Brezinski, C.: A general extrapolation algorithm. Numerische Mathematik 35, 175–187 (1980)
7. Buras, A.J., Czarnecki, A., Misiak, M., Urban, J.: Two-loop matrix element of the current-current operator in the decay $B \rightarrow X_s\gamma$. Nuclear Physics B(611), 488–502 (2001)
8. Czarnecki, A., Marciano, W.J.: Electroweak radiative corrections to $b \rightarrow s\gamma$. Phys. Rev. Lett. 81(2), 277–280 (1998)
9. Davis, P.J., Rabinowitz, P.: Methods of Numerical Integration. Academic Press, New York (1984)
10. de Doncker, E.: Numerical Integration and Asymptotic Expansions. Ph.D. thesis, Katholieke Universiteit Leuven (1980)
11. de Doncker, E., Li, S., Fujimoto, J., Shimizu, Y., Yuasa, F.: Regularization and extrapolation methods for infrared divergent loop integrals. In: Sunderam, V.S., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2005. LNCS, vol. 3514, pp. 165–171. Springer, Heidelberg (2005)

12. de Doncker, E., Li, S., Shimizu, Y., Fujimoto, J., Yuasa, F.: Numerical computation of a non-planar two-loop vertex diagram. In: LoopFest, V. (ed.) Stanford Linear Accelerator Center (2006), http://www.conf.slac.stanford.edu/loopfestv/proc/present/DEDONCKER.pdf
13. de Doncker, E., Shimizu, Y., Fujimoto, J., Yuasa, F.: Computation of loop integrals using extrapolation. Computer Physics Communications 159, 145–156 (2004)
14. de Doncker, E., Shimizu, Y., Fujimoto, J., Yuasa, F.: Computation of Feynman loop integrals. PAMM - Wiley InterScience Journal 7(1) (2007)
15. de Doncker, E., Shimizu, Y., Fujimoto, J., Yuasa, F., Cucos, L., Van Voorst, J.: Loop integration results using numerical extrapolation for a non-scalar integral. Nuclear Instuments and Methods in Physics Research Section A 539, 269–273 (2004)
16. Ferroglia, A., Passarino, G., Passera, M., Uccirati, S.: All-purpose numerical evaluation of one-loop multi-leg Feynman diagrams. Tech. rep., hep-ph/0209219
17. Ferroglia, A., Passera, M., Passarino, G., Uccirati, S.: Two-loop vertices in quantum field theory: Infrared convergent scalar configurations (2003), hep-ph/0311186
18. Fleischer, J., Tarasov, O.V.: Calculation of Feynman diagrams from their small momentum expansion. Zeitschrift für Physik C 64, 413–425 (1994)
19. Ford, W., Sidi, A.: An algorithm for the generalization of the Richardson extrapolation process. SIAM Journal on Numerical Analysis 24, 1212–1232 (1987)
20. Fujimoto, J., Shimizu, Y., Kato, K., Oyanagi, Y.: Numerical approach to one-loop integrals. Progress of Theoretical Physics 87(5), 1233–1247 (1992)
21. Fujimoto, J., Shimizu, Y., Kato, K., Oyanagi, Y.: Numerical approach to two-loop integrals. In: Proc. of the VIIth Workshop on High Energy Physics and Quantum Field Theory (1992)
22. Fujimoto, J., Ueda, T.: New implementation of the sector decomposition on FORM. In: XII Advanced Computing and Analysis Techniques in Physics Research) poS (ACAT 2008), vol. 120 (2009), ArXiv:0902.2656v1 [hep-ph]
23. Fujimoto, J., Ueda, T.: New implementation of the sector decomposition on FORM, aCAT08 talk slides (2008), http://indico.cern.ch/conferenceOtherViews.py?confId=34666&view=static&showDate=all&showSession=all&detailLevel=contribution
24. Genz, A.: The Approximate Calculation of Multidimensional Integrals using Extrapolation Methods. Ph.D. thesis, Univ. of Kent at Canterbury (1975)
25. Genz, A., Malik, A.: An imbedded family of multidimensional integration rules. SIAM J. Numer. Anal 20, 580–588 (1983)
26. Hävie, T.: Generalized Neville-type extrapolation schemes. BIT 19, 204–213 (1979)
27. HMLIB: Nucl. Instr. and Meth. A 559, 269 (2006)
28. Hooft, G., Veltman, M.: Regularization and renormalization of gauge fields. Nucl. Phys. B 44, 189 (1972)
29. Hurth, T.: Present status of inclusive rare B decays (2003), hep-ph/0212304, CERN-TH/2002-264, SLAC-PUB-9604
30. Kawabata, S.: A new version of the multi-dimensional integration and event generation package bases/spring. Computer Physics Communications 88, 309–326 (1995)
31. Kurihara, Y.: Dimensionally regularized one-loop tensor integrals with massless internal particles (2005), hep-ph/0504251 v3
32. Kurihara, Y., Kaneko, T.: Numerical contour integration for loop integrals. Computer Physics Communications 174(7), 530–539 (2006)
33. Levin, D., Sidi, A.: Two classes of non-linear transformations for accelerating the convergence of infinite integrals and series. Appl. Math. Comp. 9, 175–215 (1981)
34. Li, S.: Online Support for Multivariate Integration. PhD dissertation, Western Michigan University (December 2005)

35. Lyness, J.N.: Applications of extrapolation techniques to multidimensional quadrature of some integrand functions with a singularity. Journal of Computational Physics 20, 346–364 (1976)
36. Lyness, J.N., de Doncker, E.: On quadrature error expansions part I. Journal of Computational and Applied Mathematics 17, 131–149 (1987)
37. Lyness, J.N., de Doncker, E.: On quadrature error expansions II. The full corner singularity. Numerische Mathematik 64, 355–370 (1993)
38. Neubert, M.: Renormalization-group improved calculation of the $B \to x_s\gamma$ branching ratio. hep-ph 1(16) (2004), 0408179, CLNS 04/1885
39. Passarino, G.: An approach toward the numerical evaluation of multiloop Feynman diagrams. Nucl. Phys. B 619, 257 (2001)
40. Piessens, R., de Doncker, E., Überhuber, C.W., Kahaner, D.K.: QUADPACK, A Subroutine Package for Automatic Integration. Series in Computational Mathematics. Springer, Heidelberg (1983)
41. Shanks, D.: Non-linear transformations of divergent and slowly convergent sequences. J. Math. and Phys. 34, 1–42 (1955)
42. Sidi, A.: Convergence properties of some nonlinear sequence transformations. Math. Comp. 33, 315–326 (1979)
43. Tarasov, O.V.: An algorithm for small momentum expansion of Feynman diagrams (1995); hep-ph/9505277
44. Tkachov, F.V.: Algebraic algorithms for multiloop calculations: The first 15 years. What's next? Nucl. Phys. B 389, 309 (1997)
45. Vermaseren, J.A.M.: New features of FORM (2000), math-ph/0010025
46. Wynn, P.: On a device for computing the $e_m(s_n)$ transformation. Mathematical Tables and Aids to Computing 10, 91–96 (1956)
47. Yasui, Y., Ueda, T., de Doncker, E., Fujimoto, J., Hamaguchi, N., Ishikawa, T., Shimizu, Y., Yuasa, F.: Status reports from the grace group. In: International Colliders Workshop LCWS/ILC (2007), arXiv:0710.2957v1 [hep-ph]
48. Yuasa, F., de Doncker, E., Fujimoto, J., Hamaguchi, N., Ishikawa, T., Shimizu, Y.: Precise numerical results of IR-vertex and box integration with extrapolation. In: Proc. of the XI ACAT workshop, Advanced Computing and Analysis Techniques in physics research (2007), arXiv:0709.0777v2 [hep-ph]
49. Yuasa, F., Ishikawa, T., Fujimoto, J., Hamaguchi, N., de Doncker, E., Shimizu, Y.: Numerical evaluation of Feynman integrals by a direct computation method. In: Proc. of the XII ACAT workshop, Advanced Computing and Analysis Techniques in physics research (2008), arXiv:0904.2823

Application of Wavelet-Basis for Solution of the Fredholm Type Integral Equations*

Carlo Cattani[1] and Aleksey Kudreyko[2]

[1] diFarma, [2] Department of Mathematics and Computer Science,
University of Salerno, Via Ponte Don Melillo,
84084 Fisciano (SA), Italy
ccattani@unisa.it,
akudreyko@unisa.it

Abstract. The paper deals with the application of periodic wavelts as basis functions for solution of the Fredholm type integral equations. We examine a special case for a degenerate kernel and show multiscale solution of an integral equation for a non-degenerate kernel. The benefits of the application of periodic harmonic wavelets are discussed. The approximation error of projection of solution on the space of periodized wavelets is analytically estimated.

Keywords: Fredholm integral equations, integro-differential equation, periodized harmonic wavelets, degenerate kernels, collocation method, decomposition.

1 Introduction

The Fredholm type integral equations are often encountered in different branches of mathematical physics and applied science (e.g. [1]). It is known that the Fredholm type integral equation of the second kind can be written as follows

$$f(x) + \lambda \int_0^1 k(x,t) f(t) dt = g(x)\,, \tag{1}$$

where parameter λ is called characteristic value of the integral equation, $k(x,t)$ is an integrable function with respect to x and t $(0 \leq x \leq 1, 0 \leq t \leq 1)$. There exist hundreds types of integral equations of type (1) (e.g. see [5]). In our paper we will mainly focus our discussion on wavelet approach for solution of integral equations, where it is expected to get a periodic solution in $L^2([0;1])$ space.

This paper shows that PHW, which satisfy the axioms of the multiscale analysis [6] can be used as basis functions in solution of integral equations. The main purpose of the present chapter is to propose for numerical solution of integral equations a simple method based on PHW. The recommended technique is also

* The work of A. Kudreyko is supported by the Istituto Nazionale di Alta Matematica Francesco Severi (Rome-IT) under scholarship U 2008/000564, 21/11/2008.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 155–163, 2010.

applicable with minor changes to the Fredholm, Volterra and integro-differential equations. The error estimation shows that the accuracy of computations is very high even when the scaling parameter is small.

2 Periodized Harmonic Wavelets

In most practical applications such as image processing, data fitting, or problems involving differential equations, the space domain is a finite interval. Many of these cases can be dealt only by introducing periodized scaling functions and wavelets which we define as follows:

Definition. Let $\varphi \in L^2(\mathbb{R})$ and $\psi \in L^2(\mathbb{R})$ are the basic scaling function and the basic wavelet from MRA. For any $j, k \in \mathbb{Z}$ we define the 1 - periodic scaling function

$$\varphi^{per}_{j,k}(x) = \sum_{r=-\infty}^{\infty} \varphi_{j,k}(x+r) = 2^{j/2} \sum_{r=-\infty}^{\infty} \varphi(2^j(x+r)-k), \quad x \in \mathbb{R} \tag{2}$$

and the 1 - periodic wavelet

$$\psi^{per}_{j,k}(x) = \sum_{r=-\infty}^{\infty} \psi_{j,k}(x+r) = 2^{j/2} \sum_{r=-\infty}^{\infty} \psi(2^j(x+r)-k), \quad x \in \mathbb{R}. \tag{3}$$

It should be emphasized that many of the properties of the non-periodic scaling functions and wavelets carry over to the periodized versions restricted to the interval $[0;1]$.

Let us assume that the solution of (1) belongs to $L^2([0;1])$ and represents a periodic function. Then the unknown function $f(t)$ can be projected on the space of basis functions $\psi_{j,k}(t)$ as follows

$$\mathcal{P}_{V_N} f(t) = a_0\varphi(t) + \sum_{j=0}^{N-1}\sum_{k=0}^{2^j-1} \{a_{j,k}\psi_{j,k}(t) + a_{j,k}\overline{\psi}_{j,k}(t)\}\,, \tag{4}$$

where the bar over ψ stands for its complex conjugate, $\varphi(x)$ is the scaling function ($\varphi(x) = 1$), j is a scaling parameter, k - translation parameter and N is the approximation level, $N \in \mathbb{Z}^+$. The functions are defined by the following wavelet

$$\psi_{j,k}(t) = 2^{-j} \sum_{m=2^j}^{2^{j+1}-1} e^{2\pi i m(t-k/2^j)}\,, \tag{5}$$

which represents the set of 1-periodic functions [7,8,9] form an orthonormal basis for $L^2([0;1])$ as follows

$$\left\{1, \{\{\psi_{j,k}\}_{k=0}^{2^j-1}\}_{j=0}^{\infty}\right\}.$$

Functions $\psi_{j,k}(t)$ define periodic harmonic wavelets. The plots for several values of the scaling parameter j are shown in Fig. 1 for the selected positions k.

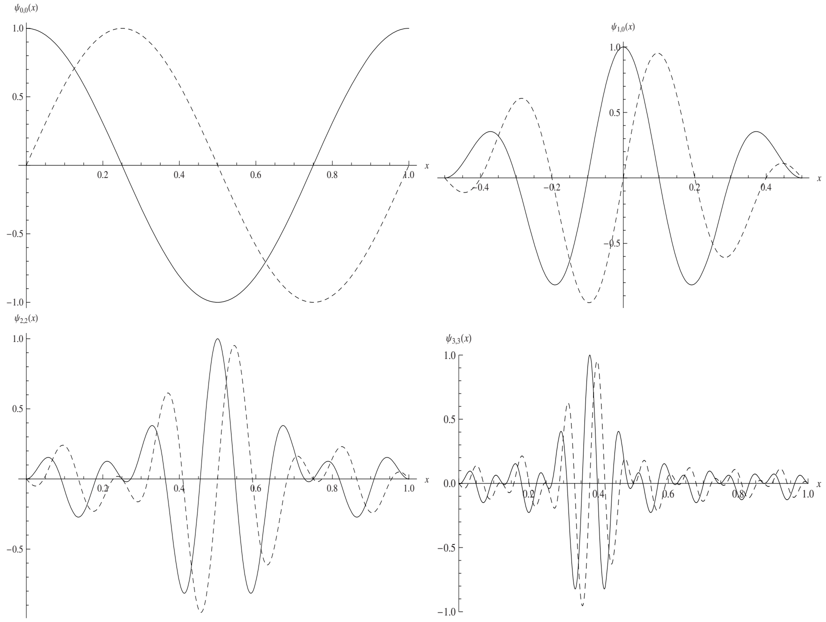


Fig. 1. Real (solid) and imaginary (dashed line) parts of the periodic harmonic wavelets $\psi_{0,0}\,(x)$, $\psi_{1,0}\,(x)$, $\psi_{2,2}\,(x)$, $\psi_{3,3}\,(x)$

The application of such basis functions in solution of partial differential equations and integral equations has several economic benefits in terms of computational costs and theoretical applications. It is shown (see e.g. [12]) that any expansion (4) of 1-periodic function from $L^2([0;1])$ converges to the corresponding function. Thus, if we define wavelet-coefficients, an analytical expansion can be obtained, which is much better than approximation by the Haar or Daubechies wavelets.

Some of the theoretical applications of such basis are considered in the proceeding chapter.

3 Collocation Method for Integral Equations with Degenerate Kernels

This section is intended to show how can be solved an integral equation with a degenerate kernel by using periodized wavelets and basic properties of wavelet expansion. If the projection of solution on the space of periodic harmonic wavelets V_N represents approximation of an unknown function $f(t)$, then we can substitute (4) into (1), and we find

$$a_0 + \sum_{j,k} \{a_{j,k}\psi_{j,k}(x) + \widetilde{a}_{j,k}\overline{\psi}_{j,k}(x)\} + a_0 \int_0^1 k(x,t)dt + \underbrace{\int_0^1 k(x,t)\left(\sum_{j,k} a_{j,k}\psi_{j,k}(t)\right)dt}_{a} + \underbrace{\int_0^1 k(x,t)\left(\sum_{j,k} \widetilde{a}_{j,k}\overline{\psi}_{j,k}(t)\right)dt}_{b} = g(x)\,, \tag{6}$$

where $\sum\limits_{j,k} = \sum\limits_{j=0}^{N-1}\sum\limits_{k=0}^{2^j-1}$. The obtained equation with respect to $\{a_0, a_{j,k}\}$ can be solved for example by collocation method. To gain the full benefit of the approach we take, let consider integrals marked as "a" and "b".

Indeed, integrals "a" and "b" are similar. Therefore, we will consider only one integral (e.g. "a") and spread our results on integral "b". According to the theorem of decay of wavelet coefficients [11], terms "a" and "b" satisfy the criterion for interchanging summation and integration, i.e. $\sum |a_{j,k}| < \infty$. Let us also assume that kernel $k(x,t)$ is degenerate, which means

$$k(x,t) = \sum_l h_l(x)p_l(t)\,.$$

Then we can write

$$\int_0^1 k(x,t)\left(\sum_{j,k} a_{j,k}\psi_{j,k}(t)\right)dt = \sum_{j=0}^{N-1}\sum_{k=0}^{2^j-1} a_{j,k}\sum_{l=1}^{n} h_l(x)\int_0^1 p_l(t)\psi_{j,k}(t)dt\,. \tag{7}$$

Integrals $\int\limits_0^1 p_l(t)\psi_{j,k}(t)dt$ can be computed explicitely, and equation (6) can be easily reduced to a system of linear algebraic equations by denoting collocation points as follows

$$0 \leq x_1 < x_2 < \ldots \leq 1\,.$$

Let us note that the idea of such representation of an unknown function in integral equation is not new. However, the idea of application of periodic harmonic wavelets for solution of such equations is new. The advantage of such approach is that the approximation error of periodized wavelets is low.

4 Approximation Properties of Multiresolution Spaces

Let us now consider the approximation error for the periodic wavelets. Let $f(x) \in L^2([0;1])$ and assume that its periodic expansion (4) is P times differentiable everywhere. Denote the approximation error as follows

$$e_N^{per}(x) = f(x) - \mathcal{P}_{V_N}f(x)\,, \quad x \in [0;1] \tag{8}$$

where $\mathcal{P}_{V_N} f(x)$ is the orthogonal projection of $f(x)$ onto the space of PHW. The symbol "*per*" over e_N assumes that the error is a periodic function. The derivation of the value of $e_N^{per}(x)$ is presented in the following theorem.

Theorem 1. *The approximation error* (8) *is bounded by the exponential decay* $|e_N^{per}(x)| = \mathcal{O}(2^{-NP})$.

Proof. Using the wavelet periodic expansion (4), we find that

$$\mathcal{P}_{V_N} f(x) = \sum_{k=0}^{\infty} a_{\varphi,k}\varphi(x-k) + \sum_{j=0}^{N-1}\sum_{k=0}^{2^j-1} a_{j,k}\psi_{j,k}(x)\,. \tag{9}$$

At any given scale, the projection of the function on the subspace of wavelets of the certain scale approaches to the function as the number of zero wavelet moments P tends to infinity, i.e. $N \to \infty$, we get $f(x)$ itself:

$$f(x) = \sum_{k=0}^{\infty} a_{\varphi,k}\varphi(x-k) + \sum_{j=0}^{\infty}\sum_{k=0}^{\infty} a_{j,k}\psi_{j,k}(x)\,. \tag{10}$$

Then, by subtracting (9) from (10), we obtain an expression for the error e_N^{per} in terms of the wavelets at scales $j \geq N$:

$$e_N^{per}(x) = \sum_{j=N}^{\infty}\sum_{k=0}^{2^j-1} a_{j,k}\psi_{j,k}(x)\,. \tag{11}$$

Define

$$C_\psi = \max_{x\in I_{j,k}} |\psi(2^j x - k)| = \max_{y\in[0,D-1]} |\psi(y)|\,.$$

Since $\max_{x\in I_{j,k}} |\psi_{j,k}(x)| = 2^{j/2}C_\psi$ and according to the Theorem of decay of wavelet coefficients [11], it is

$$|a_{j,k}\psi_{j,k}(x)| \leq C_P 2^{-jP} \max_{\xi\in I_{j,k}} |f^{(P)}(\xi)| C_\psi\,.$$

Recall that

$$\operatorname{supp}(\psi_{j,k}) = I_{j,k} = \left[\frac{k}{2^j}; \frac{k+D-1}{2^j}\right]\,.$$

Hence, there are at most $D-1$ intervals $I_{j,k}$ containing a given value of x. Thus, for any x only $D-1$ terms in the inner summation in (11) are nonzero. Let I_j be a union of all these intervals, i.e.

$$I_j(x) = \bigcup_{\{l:x\in I_{j,l}\}} I_{j,l}$$

and let

$$\mu_j^P(x) = \max_{\xi\in I_j(x)} |f^P(\xi)|\,.$$

Then we can find a common bound for all terms in the inner sum:

$$\sum_{k=-\infty}^{\infty} |a_{j,k}\psi_{j,k}| \leq C_\psi C_P 2^{-jP}(D-1)\mu_j^P(x)\,.$$

The outer sum over j can be evaluated using the fact that

$$\mu_N^P(x) \geq \mu_{N+1}^P(x) \geq \mu_{N+2}^P(x) \geq \dots$$

and we establish the bound

$$\begin{aligned} |e_N^{per}(x)| &\leq C_\psi C_P(D-1)\mu_N^P(x)\sum_{j=N}^{\infty} 2^{-jP} \\ &= C_\psi C_P(D-1)\mu_N^P(x)\frac{2^{-NP}}{1-2^{-P}}\,. \end{aligned}$$

Thus, we see that for an arbitrary, but fixed x, the approximation error will be bounded as follows:

$$|e_N^{per}(x)| = \mathcal{O}(2^{-NP})\,,$$

where $\mathcal{O}$ only denotes an upper bound. This is an exponential decay with respect to the resolution N. Furthermore the greater number of vanishing moments P of a periodic wavelet increases the rate of the decay.

Let us compare the approximation error of wavelets with the error of the Fourier approximation for N terms. In order to do this, we need to introduce a smooth function of the order q.

Definition. A smooth function is a function that has continuous derivatives up to some desired order q over some domain. A function can therefore be said to be smooth over a restricted interval such as $[a;b]$.

According to [13], we can find that the approximation error of the Fourier series is

$$|e^F(q,N)| = \max_{a\leq x\leq b} |F(N,x) - f(x)| = \mathcal{O}(N^{-q-0.5}).$$

This is also an exponential decay with respect to the number of terms in the series and the level of smoothness of a function. In order to give a more detailed comparison of these two methods, it is necessary to consider specific examples.

5 Multiscale Solution of the Fredholm Integral Equation

This section illustrates our approach and the property of periodic harmonic wavelets to construct multiresolution analysis. Let us solve the following integral equation.

Example 1

$$\int_0^1 f(t)e^{t\sin x}dt = -\frac{(1-e^{\sin x})(\sin^3 x + 16\pi^2\sin x - 8\pi^3 - 2\pi\sin^2 x)}{(\sin^2 x + 8\pi^2)^2 + 4\pi^2\sin^2 x}\,. \qquad (12)$$

It should be noted that such special form of the equation (12) is taken only to demonstrate the applicability of periodized hatmonic wavelets and multiresolution analysis. In fact, there could be any 1-periodic function from the space $L^2([0;1])$. Now let us find the solution on the lowest approximation level, i.e. $N = 1$. Then, taking into account (4) and (5), the projection of the solution on the space of periodic harmonic wavelets will be

$$\mathcal{P}_{V_1} f(t) = a_0 + a_{0,0} e^{2\pi i t} + \widetilde{a}_{0,0} e^{-2\pi i t} .$$

If the expression for $\mathcal{P}_{V_1} f(t)$ is inserted into (12), we obtain

$$\int\limits_0^1 (a_0 + a_{0,0} e^{2\pi i t} + \widetilde{a}_{0,0} e^{-2\pi i t}) e^{t \sin x} dt = X ,$$

where X is the right hand side of (12). Denote the collocation points as follows $x_1 = 0$, $x_2 = \pi/6$, $x_3 = \pi/4$. Then we will get a system of linear algebraic equations with respect to wavelet-coefficients

$$\int\limits_0^1 (a_0 + a_{0,0} e^{2\pi i t} + \widetilde{a}_{0,0} e^{-2\pi i t}) e^{t \sin x_l} dt = X_l ,$$

where $l = 1, 2, 3$. If we assume that $f(t)$ is a real valued function, then we can take advantage of the equality $\overline{a}_{j,k} = \widetilde{a}_{j,k}$ [8]. The solution of such system yields us the following coefficients: $a_0 = 0, a_{0,0} = 0.487, \widetilde{a}_{0,0} = 0.487$ and the corresponding plot is presented in Fig. 2 (dashed line). In the case if we assume that $N = 2$, we find

$$\begin{aligned}\mathcal{P}_{V_2} f(t) =& a_0 + a_{0,0}\psi_{0,0}(t) + \widetilde{a}_{0,0}\psi_{0,0}(t) \\ & + a_{1,0}\psi_{1,0}(t) + \widetilde{a}_{1,0}\overline{\psi}_{1,0}(t) + a_{1,1}\psi_{1,1}(t) + \widetilde{a}_{1,1}\overline{\psi}_{1,1}(t).\end{aligned}$$

And the corresponding choice of collocation points will also give us a system of algebraic equations with respect to wavelet coefficients. It can be shown that these coefficients are $a_0 = 0, a_{0,0} = \widetilde{a}_{0,0} = 0.5, a_{1,0} = -i/8, \widetilde{a}_{1,0} = i/8, a_{1,1} = -i/8, \widetilde{a}_{1,1} = i/8$. And the plot for $N = 2$ is presented in Fig. 2 by solid line. It is obvious that the first approximation perpesents the raw approximation of the second level. The projection of solution on the second level of approximation represents the exact solution of integral equation (12).

If we have continued our computations for $N = 3$ etc, the other wavelet coefficients would eventually be zeros.

Example 2. Consider another type of problem, i.e. an integro-differential equation [4]

$$\begin{cases} y'(x) = \int\limits_0^1 \sin(2\pi t + 4\pi x) y(t) dt - \cos 2\pi x(1 + \sin 2\pi x) - 2\pi \sin 2\pi x \\ y(0) = 1 . \end{cases} \tag{13}$$

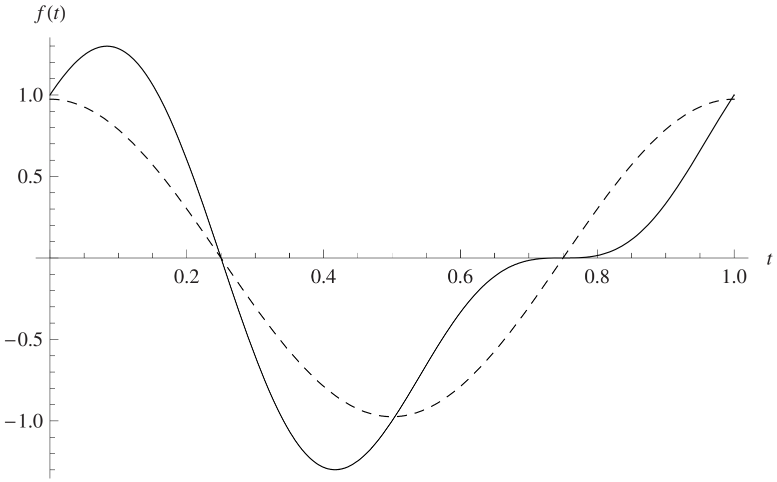


Fig. 2. Multiscale solution of equation (12) at $N = 1$- dashed line and $N = 2$ solid line

Similar to the previous example, we can assume that the solution can be represented in terms of periodic wavelets, i.e.

$$\mathcal{P}_{V_1} f(t) = a_0 + a_{0,0} e^{2\pi i t} + \widetilde{a}_{0,0} e^{-2\pi i t} \, .$$

Then the corresponding choice of collocation points gives us a system of linear algebraic equations with respect to $a_0, a_{0,0}, \widetilde{a}_{0,0}$. The values of wavelet coefficients are $a_0 = 0; a_{0,0} = \widetilde{a}_{0,0} = 0, 5$.

Thus, our basis functions allow to expand an unknown function in terms of periodic wavelets. This property can be used in solution of PDEs [2,10], integral equations [3] and integro-differential equations.

6 Concluding Remarks

Integral equations, which describe mathematical models can be treated by many analytical and numerical methods. The approach that we study yields analytical approximation for a certain class of problems. We have shown that our basis functions can be applied for solution of the Fredholm type integral equations. Such application of periodic wavelets yields the projection of solution on their space and can be presented at different scales. There are three important facts to note about the wavelet approximation.

1. The good resolution of the discontinuity is a consequence of the large wavelet coefficients appearing at the fine scales.
2. The fact that the error is restricted to a small neighbourhood of the discontinuity is the result of the "locality" of wavelets. The behaviour of $f(x)$ at one location affects only the coefficients of wavelets close to that location.
3. Most of the linear part of $f(x)$ is represented exactly.

We conclude by pointing out what we see as the main obstacles for obtaining truly feasible and competitive wavelet-based solvers for integral equations. Part of the problem is the fact that there is always a finest level inherent to all wavelet approaches. Future research could be directed at methods for non-periodic boundary conditions.

Much of the research in this field is in progress and it is too early to say whether the problems mentioned above will be solved. However, there is no doubt that wavelet analysis has earned its place as an important alternative to Fourier analysis, only the scope of applicability remains to be settled.

The approach, that we propose does not claim to be a universal, and it is not considered as a good or the best, we only made an attempt to extend the borders of the application of PWH. Unfortunately, in the theory that we develop PHW do not approximate non-periodic functions nearby bounds and the choice of a wavelet for the analysis of data represents more an “art” than a routine operation.

References

1. Cassell, J.S., Williams, M.M.R.: An integral equation arising in neutron transport theory. Annals of Nuclear Energy 30, 1009–1031 (2003)
2. Cattani, C.: Multiscale Analysis of wave propagation in composite materials. Mathematical Modelling and Analysis 8(4), 267–282 (2003)
3. Cattani, C., Kudreyko, A.: Application of periodized harmonic wavelets towards solution of eigenvalue problems for integral equations. Mathematical Problems in Engineering (2010) (In press)
4. Danfu, H., Xufeng, S.: Numerical solution of integro-differential equations by using CAS wavelet operational matrix of integration. Applied Mathematics and Computation 194, 460–466 (2007)
5. Polyanin, A.D., Manzhirov, A.V.: Handbook of integral equations, 796 p. CRC Press, London (1998)
6. Daubechies, I.: Ten Lectures on Wavelets, 377 p. SIAM, Philadelphia (1992)
7. Newland, D.E.: Harmonic wavelets in vibrations and acoustics. The Royal Society 357, 2607–2625 (1999)
8. Newland, D.E.: Harmonic wavelet analysis. The Royal Society 443, 203–225 (1993)
9. Newland, D.E.: An Introduction to Random Vibrations. In: Spectral & Wavelet Analysis, 3rd edn. (1993)
10. Muniandy, S.V., Moroz, I.M.: Galerkin modeling of the Burgers equation using harmonic wavelets. Physics Letters A 235, 352–356 (1997)
11. Mallat, S.: A Wavelet Tour of Signal Processing. In: École Polytechnique, Paris, 2nd edn. Academic Press, London (1990)
12. Morita, T.: Expansion in Harmonic Wavelets of a Periodic Function. Interdiscipl. Inform. Sci. 3(1), 5–12 (1997)
13. Miroshnichenko, G.P., Petrashen, A.G.: Numerical methods, Saint-Petersburg, 120 p. (2008)
14. Bakhoum, E., Toma, C.: Mathematical Transform of Travelling-Wave Equations and Phase Aspects of Quantum Interaction. Mathematical Problems in Engineering 2010, Article ID 695208(2010)
15. Toma, G.: Specific Differential Equations for Generating Pulse Sequences. Mathematical Problems in Engineering 2010, Article ID 324818 (2010)

Fractal Patterns in Prime Numbers Distribution

Carlo Cattani

diFarma, University of Salerno,
Via Ponte Don Melillo, 84084 Fisciano (SA) Italy
ccattani@unisa.it

Abstract. One of the main tasks in the analysis of prime numbers distribution is to single out hidden rules and regular features like periodicity, typical patterns, trends, etc. The existence of fractal shapes, patterns and symmetries in prime numbers distribution are discussed.

Keywords: Fractals, fractal dimension, random walk, prime numbers, Riemann hypothesis.

1 Introduction

Some open questions in primes distribution [12,13,16] analysis are the understanding of the underlying rule, to find an organization principle, to discover some kind of order (symmetries) or hidden structures (patches or regular patterns) and the existence of localized periodicities, correlation, complexity, etc. [1,17,13]. The main problem is to find out (if any) some kind of mathematical rules or meaningful statistics, in the primes distribution within the positive integers greater than one: $\mathbb{Z}_{>1}$.

From mathematical point a view the primes distribution analysis is carried out on a very large sequences of integers. Within these large sequences the primes distribution looks like a random sequence, from where it seems to be quite impossible to single out any kind of correlation.

A fundamental problem in primes analysis is wether there exist a statistical correlation (or long range-correlation) in a suitable representation of primes within the natural numbers. Correlation in a sequence can be roughly linked with the concept of dependence, in a statistical sense, of elements which are far away from each other. Due to this there follows the characterizing $1/f$ power law decay.

The power law for long-range correlations is a measure of the scaling law, showing the existence of self-similar structures similar to the physics of fractals. The long-range correlation, which can be detected by the autocorrelation function, implies the scale independence (scale invariance) which is typical of fractals. The autocorrelation is also used for measuring linear dependence and periodicity. The existence of patchiness and correlation would imply some important understanding of primes. It can be observed that the source for long-range correlation might be linked with existence of patchiness in primes. The identification of these patches could be the key point for understanding the large scale structure of primes distribution.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 164–176, 2010.

2 Preliminary Remarks on Primes

Given positive integers a and n we say that a divides n (n is divisible by a) if and only if there exists a positive integer b such that

$$n = ab \; .$$

In other words, given two positive integers $a, \; n \in \mathbb{N}$

$$a|n \Leftrightarrow \exists \, b \in \mathbb{N} : n = ab \; ,$$

in this case it is said that a is a divisor of n (or a divides n).

Let

$$Div(n) \overset{\text{def}}{=} \{m \in \mathbb{N} : m|n\}$$

be the set of positive divisors of n and $|Div(n)|$ the cardinality of the set, we say that a positive integer p is a prime if $|Div(p)| = 2$, so that an integer p is a prime if its only positive divisors are 1 and p. The set of all primes is

$$\mathbb{P} \overset{\text{def}}{=} \{p \in \mathbb{N} : |Div(p)| = 2\}$$

and a fundamental elementary theorem of Arithmetic states that every integer larger than 1 can be expressed as a product of primes, so that any positive integer has a unique prime factorization (up to a suitable ordering), and primes play the role of atoms for the positive integers.

It was known, already by Euclid's time, that the number of primes is infinite, i.e. $|\mathbb{P}| = \infty$, however it is still unknown how they are distributed within $\mathbb{N}$.

If we define the counting function, $\pi(x) : \mathbb{R} \Rightarrow \mathbb{N}$, as

$$\pi(x) = |\mathbb{P}_x| \quad , \quad \mathbb{P}_x \overset{\text{def}}{=} \{p \in \mathbb{P} : p \leq x\} \quad , \quad \mathbb{P}_x \subseteq \mathbb{P}$$

it has been conjectured by Gauss that $\pi(x)$ asymptotically tends to $x/\log x$, i.e.

$$\pi(x) \sim x/\log x \tag{1}$$

so that the prime number theorem

$$\lim_{x\to\infty} \frac{\pi(x)}{x/\log x} = 1$$

holds.

2.1 The Riemann Zeta Function

The Riemann Zeta function is defined as

$$\zeta(s) \overset{\text{def}}{=} \sum_{n=1}^{\infty} \frac{1}{n^s} \quad , \quad (n \in \mathbb{N} \, , s \in \mathbb{C}) \tag{2}$$

Let us take

$$s = y + ix$$

It can be seen that when s is a pure imaginary number ($y = 0\ ,\ s = i\,x$), it is

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^{ix}} \quad , \quad (n \in \mathbb{N}\ ,\ x \in \mathbb{R})\ .$$

The absolute value $|\zeta(ix)|$ is a decaying function only in a finite interval (see e.g. Fig. 1)

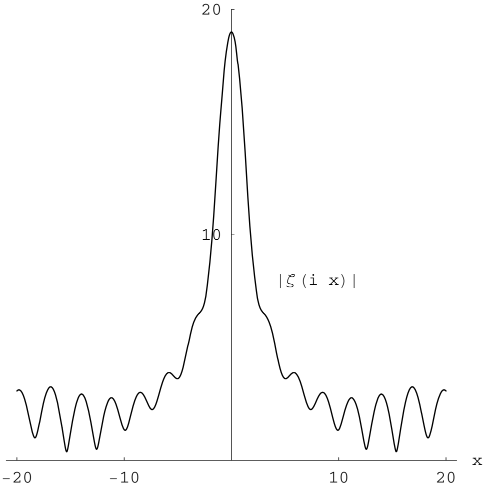


Fig. 1. $|\zeta(ix)|$ when $(n \leq 20)$

In particular, the absolute value of the function

$$\zeta^N(x) \overset{\text{def}}{=} \sum_{n=1}^{N} \frac{1}{n^{ix}} \quad , \quad (n \in \mathbb{N},\ x \in \mathbb{R}) \tag{3}$$

decays in the interval $\left[-\frac{N}{2}, \frac{N}{2}\right]$ and oscillates, with bounded amplitude, elsewhere.

For a fixed N both the real and imaginary part (Fig. 2) of $\zeta^N(x)$ show some slow decay to zero.

By a direct computation it can be seen that

$$\int_{-a}^{a} \Re(\zeta^N(x))\mathrm{d}x = \sum_{n=2}^{N} 2\frac{\sin(a \log n)}{\log n} \quad , \quad (N \geq 2)$$

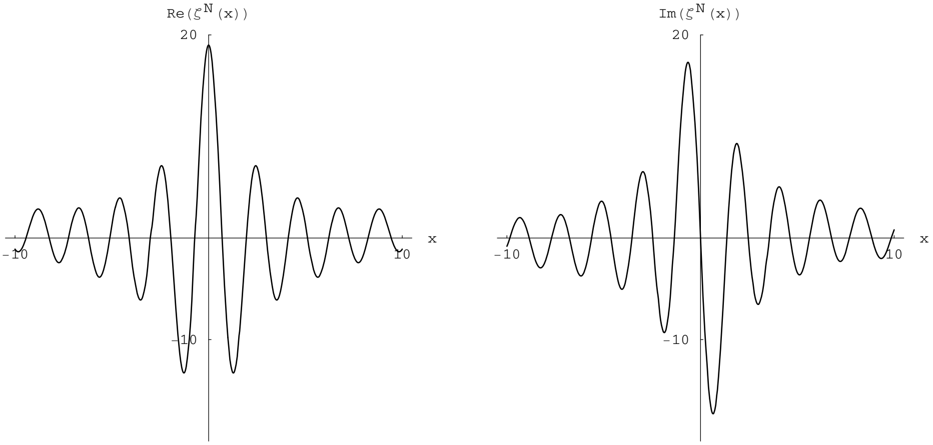


Fig. 2. $\Re[\zeta^N(x)]$ and $\Im[\zeta^N(x)]$ when $(n \leq N = 20)$

and

$$\int_{-ai}^{ai} \Im(\zeta^N(x))\mathrm{d}x = 0 \quad , \quad (N \geq 1) \; .$$

It is also

$$\int_{-a}^{a} \Re(\zeta^N(x))\Im(\zeta^N(x))\mathrm{d}x = 0 \quad , \quad (N \geq 1)$$

so that the real and imaginary part of the function (3) are (somehow) orthogonal in the interval $[-a,\ a]$.

It is important to notice that

$$\zeta(s) \overset{(2)}{=} \sum_{n=1}^{\infty} \frac{1}{n^y} \frac{1}{e^{ix \log n}}$$

so that

$$|\zeta(s)| \leq \sum_{n=1}^{\infty} \frac{1}{n^y} \quad , \quad y = \Re(s) \; .$$

There follows that, when $y = \Re(s) \in (0, 1)$, then $\zeta(s)$ is a bounded function.

2.2 Euler Identity

It has been shown by Euler that the following identity

$$\zeta(s) \overset{\text{def}}{=} \sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_{p \in \mathbb{P}} \left(1 - \frac{1}{p^s}\right)^{-1} \quad , \quad (s \in \mathbb{C}) \tag{4}$$

holds, when the product (on r.h.s.) is taken over the prime numbers p.

Some values of $\zeta(s)$ have been computed, for real values of s, for instance it is

$$\zeta(2) = \frac{\pi^2}{6} \cong 1.64 \quad , \quad \zeta(4) = \frac{\pi^4}{90} \; .$$

In general, when we consider complex value variables, for the Riemann function (2) we have the functional equation

$$\pi^{-s/2}\Gamma(\frac{s}{2})\zeta(s) = \pi^{-(1-s)/2}\Gamma((1-s)/2)\zeta(1-s)$$

where $\Gamma(s)$ is the Gamma-function.

If we define

$$\xi(s) \overset{\text{def}}{=} \pi^{-s/2}\Gamma(\frac{s}{2})\zeta(s)$$

the above functional identity it is known in the nice form

$$\xi(s) = \xi(1-s)$$

which shows that there is a point symmetry with respect to $s = \dfrac{1}{2}$.

Concerning the zeroes of the Riemann function, it should be noticed from (4) that when $y = \Re(s) > 1$ none of the terms of the product (on the r.h.s) are zero and for this reason

$$\zeta(s) \neq 0 \quad , \quad \Re(s) > 1 \quad .$$

Therefore the trivial zeroes are on the half-plane $\Re(s) < 1$, the non-trivial zeroes are symmetrically distributed with respect to line $\Re(s) = \dfrac{1}{2}$

Riemann conjectured that they are all located on the critical line

$$\Re(s) = \frac{1}{2}$$

thus giving some constraints about the distribution of primes.

3 Indicator Function

In order to single out some patterns in the primes distribution, we will consider in this section the indicator matrix that has been successfully applied to the analysis of DNA sequences [6].

Let $\mathcal{S} \subset \mathbb{N}$ be a finite ordered subsequence of the positive integers. We can define a sequence of primes localization by the boolean operator

$$v : \mathcal{S} \rightarrow \{0\,,1\}$$

such that for $h \in \mathcal{S}$,

$$v_h \overset{\text{def}}{=} v(h) = \begin{cases} 1 & \text{if } \{h \in \mathbb{P}\} = \text{ TRUE} \\ 0 & \text{if } \{h \in \mathbb{P}\} = \text{ FALSE} \end{cases} \; . \tag{5}$$

The indicator function is the map

$$u : \mathcal{S} \times \mathcal{S} \rightarrow \{0\ ,1\}$$

such that for $h \in \mathcal{S}$, $k \in \mathcal{S}$

$$u_{hk} \overset{\text{def}}{=} u(h,\ k) = \begin{cases} 1 & \text{if } \{h \in \mathbb{P}\} \wedge \{k \in \mathbb{P}\} = \text{ TRUE} \\ 0 & \text{if } \{h \in \mathbb{P}\} \wedge \{k \in \mathbb{P}\} = \text{ FALSE} \end{cases} . \tag{6}$$

According to (6), the indicator of a N-length sequence can be easily represented by the $N \times N$ sparse symmetric matrix $\{u_{hk}\}$ of binary values $\{0, 1\}$, as the following table

⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋰
11	1	1	0	1	0	1	0	0	0	1	…
10	0	0	0	0	0	0	0	0	0	0	…
9	0	0	0	0	0	0	0	0	0	0	…
8	0	0	0	0	0	0	0	0	0	0	…
7	1	1	0	1	0	1	0	0	0	1	…
6	0	0	0	0	0	0	0	0	0	0	…
5	1	1	0	1	0	1	0	0	0	1	…
4	0	0	0	0	0	0	0	0	0	0	…
3	1	1	0	1	0	1	0	0	0	1	…
2	1	1	0	1	0	1	0	0	0	1	…
u_{hk}	2	3	4	5	6	7	8	9	10	11	…

where both on bottom and on left there is the sequence $\mathcal{S}$, and the composition table is done according to the indicator values u_{hk}.

Matrix (6) can be plotted in 2 dimensions (Fig. 3) by putting a dot where $u_{hk} = 1$ and white spot when $u_{hk} = 0$.

It can be seen from Fig. 3, that

1. there are some motifs which are repeated at different scales like in a fractal;
2. empty spaces are more distributed than filled spaces, in the sense that the matrix u_{hk} is a sparse matrix (having more zeroes than ones);
3. it seems that there are some square-like islands where black spots are more concentrated.

3.1 Global Fractal Estimate by the Correlation Matrix

Let $p(x)$, $x \in \mathbb{R}$ be the probability to find a prime at the distance x. For large values of x it is

$$p(x) \cong \frac{\pi(x)}{x} \overset{(1)}{=} \frac{1}{\log x}$$

So that, according to Gauss conjecture, the possibility to find some primes is vanishing for higher values of x. Let us compute the number of 1 in the minor $2^m - n \times 2^m - n$ of the indicator matrix $2^m \times 2^m$. We can see (Fig. 4) that for

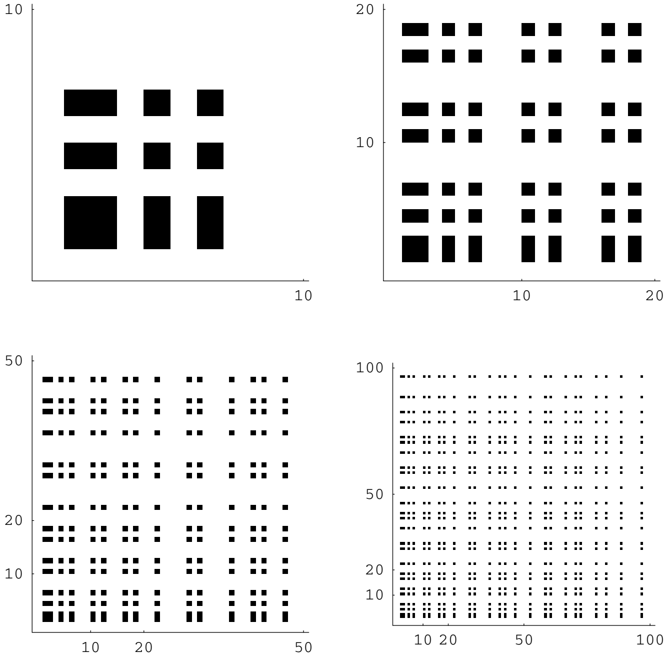


Fig. 3. Indicator matrix with $n \leq 10,\ n \leq 20$ (top) and $n \leq 50,\ n \leq 100$ (bottom)

higher values of n we find much more primes, but this probability reduces to zero beyond 2^24.

If we count the number of 1 in the $n \times n$ indicator matrix as a function of n we have the plot of frequencies (Fig. 5) which is similar to a Cantor function, thus suggesting us that the primes are distributed (within the indicator matrix) as fractals.

By using the indicator matrix it is possible to give a simple formula which enables us to estimate the fractal dimension as the average of the number $p(n)$ of 1 in the randomly taken $n \times n$ minors of the $N \times N$ correlation matrix u_{hk}

$$D = \frac{1}{N} \sum_{n=2}^{N} \frac{\log p(n)}{\log n} \quad . \tag{7}$$

By a direct computation we obtain that the fractal dimension of the primes distribution in the matrix is roughly $\sqrt{2}$. If we count the number of zeroes and the number of ones in the indicator matrix we have as a ratio

$$\frac{\log[1 - p(n)]}{\log p(n)} \cong \frac{1}{\sqrt{2}}$$

which concides with

$$\frac{\log[1 - \pi(x)]}{\log \pi(x)} \quad .$$

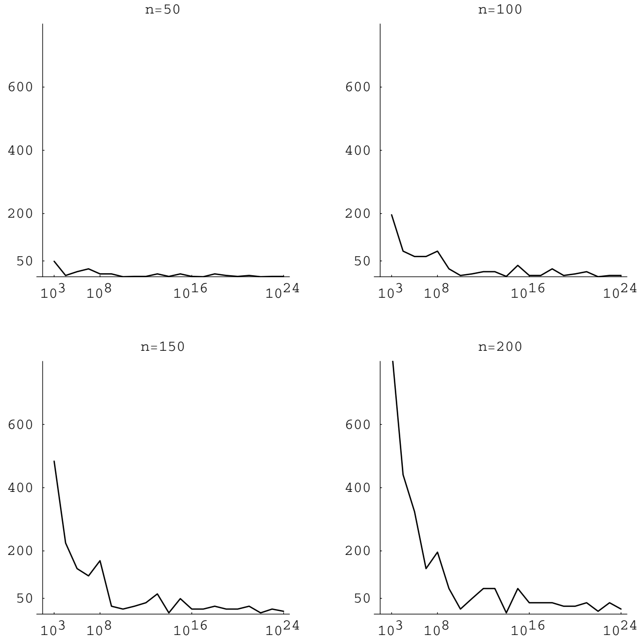


Fig. 4. Frequencies of the nonvanishing terms in the minors $2^m - n \times 2^m - n$ of the Indicator matrix for different values of n

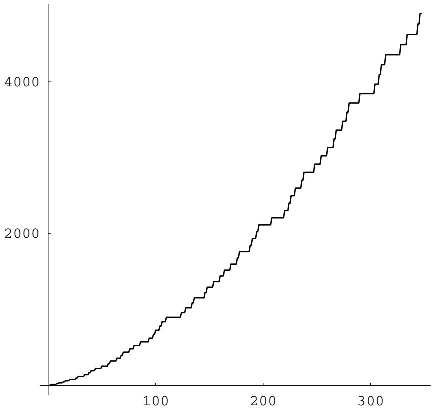


Fig. 5. Frequencies of 1 in the indicator matrix ($n \leq 350$)

4 Statistical Correlations in Primes Distribution

In order to understand the primes distribution we discuss in this section the existence of correlations.

4.1 Long-Range Correlation

The most popular techniques for measuring correlations in large time series (such as DNA) are

- the direct computation of the correlation function
- analysis of variance [14] later improved by the detrended fluctuation analysis [15],
- power spectrum method [11,18]
- mutual information function [10]
- wavelets method [2,3].

For a given sequence $\{Y_0,\ Y_1,\ \dots,\ Y_{N-1}\}$ the variance is

$$\sigma^2 \overset{\text{def}}{=} \frac{1}{N}\sum_{i=0}^{N-1} Y_i^2 - \left(\frac{1}{N}\sum_{i=0}^{N-1} Y_i\right)^2 \tag{8}$$

and the variance at the distance $N-k$

$$\sigma_k^2 \overset{\text{def}}{=} \frac{1}{N-k}\sum_{i=0}^{N-k-1} Y_i^2 - \left(\frac{1}{N-k}\sum_{i=0}^{N-k-1} Y_i\right)^2 \tag{9}$$

From the variance follows immediately the standard deviation

$$\sigma = \sqrt{\sigma^2} \tag{10}$$

The autocorrelation at the distance $k,\ (k=0,\dots,N-1)$ is the sequence (see e.g. [5])

$$c_k \overset{\text{def}}{=} \frac{1}{\sigma^2}\left(\frac{1}{N-k}\sum_{i=0}^{N-k-1} Y_i Y_{i+k} - \frac{1}{(N-k)^2}\sum_{i=0}^{N-k-1} Y_i \sum_{i=0}^{N-k-1} Y_{i+k}\right) \tag{11}$$

with $k=0,\dots,N-1$.

A simplified definition of correlation, in the fragment $F-N$ has been given [8] as follows:

$$c_k \overset{\text{def}}{=} \sum_{i=F}^{N-1-k} \frac{1}{N-F-k} u_{x_i}(x_{i+1+k})$$

with the indicator given by (6).

The power spectrum can be computed as the Fourier transform of c_k:

$$S_k \overset{\text{def}}{=} \widehat{c}_k = \sum_{n=0}^{N-1} c_n e^{-2\pi ink/N} \ .$$

If $c_k = 0$ there is no linear correlation, $c_k > 0$ means that there is a strong (linear) correlation (anticorrelation when $c_k < 0$), while $c_0 = 1$ doesn't give any information about correlations. A true random process has a vanishing correlation $c_h = \delta_{0h}$ and its power spectrum S_h is constant. Its integral gives the Brownian motion (random walk) whose power spectrum is proportional to $1/k^2$.

It is known [5,19] that to have accuracy and to avoid statistical fluctuations in the computation of the autocorrelation function a long sequence is needed. The statistical fluctuation is $\epsilon = \dfrac{1}{\sqrt{N}}$ so that the autocorrelation is measured by $c_k \pm \dfrac{1}{\sqrt{N}}$. Therefore shorter is the sequence and larger is its fluctuation.

Moreover, there are several critical comments on the direct measure of correlation:

- different sequences may exhibit different correlation functions
- correlation function obtained for the whole sequence may be different for a subsequence

4.2 Power Spectrum

Let $\{Y_n\}_{n=0,\dots,N-1}$ be a given series, the discrete Fourier series is the sequence

$$\widehat{Y}_s = \frac{1}{N}\sum_{n=0}^{N-1} Y_n e^{-2\pi ins/N} \quad , \quad s = 0, \dots, N-1 \; .$$

The power spectrum of the sequence $\{Y_n\}_{n=0,\dots,N-1}$, that is the mean square fluctuation, is defined as [7]

$$S_k \overset{\text{def}}{=} \sum_{s=0}^{k-1} \left|\widehat{Y}_s\right|^2 \tag{12}$$

The power spectrum of a stationary sequence, gives an indirect measure of the autocorrelation. A long-range correlation, can be detected if the fluctuations can be described by a power law so that

$$S_k \cong \alpha \frac{k}{\max\limits_{1\leq k\leq k_{max}} [\alpha k]} \quad , \quad 1 \leq k \leq k_{\max}$$

with $\alpha > \dfrac{1}{2}$.

The fluctuation exponent α, with its values, characterizes a sequence as

1. anti-correlated: $\alpha < 1/2$
2. uncorrelated (white noise): $\alpha \cong 1/2$
3. correlated (long range correlated): $\alpha > 1/2$
4. $1/f$ noise: $\alpha \cong 1$
5. non-stationary, random-walk like: $\alpha > 1$
6. Brownian noise: $\alpha \cong 3/2$.

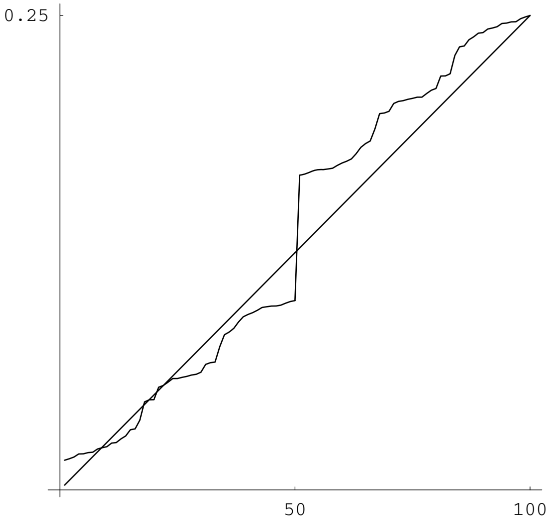


Fig. 6. Power spectrum for primes sequence $v(h)$ with $h \leq 100$ with the corresponding least square linear fit $\alpha \cong 0.002$

For the primes sequence indicator (5) it is $\alpha = 0.002$. so that the distribution can be considered anti-correlated.

If we compute the walk on $v(k)$:

$$\sum_{k=1}^{N} \frac{v(k)}{k}$$

we can see that the walk is decaying to a constant value (see Fig. 7, left) and its power spectrum is shown in Fig. 7, right.

4.3 Complexity

The existence of repeating motifs, periodicity and patchiness can be considered as a simple behavior of sequence. While non-repetitiveness or singularity might be taken as a characteristic feature of complexity. In order to have a measure of complexity, for an n-lenght sequence, it has been proposed [4] the following

$$K = \log \Omega^{1/n}$$

with

$$\Omega = \frac{n!}{\pi(n)!}$$

By using a sliding n-window [4] over the full sequence $v(k)$ one can visualize the distribution of complexity on partial fragment of the sequence. It is interesting

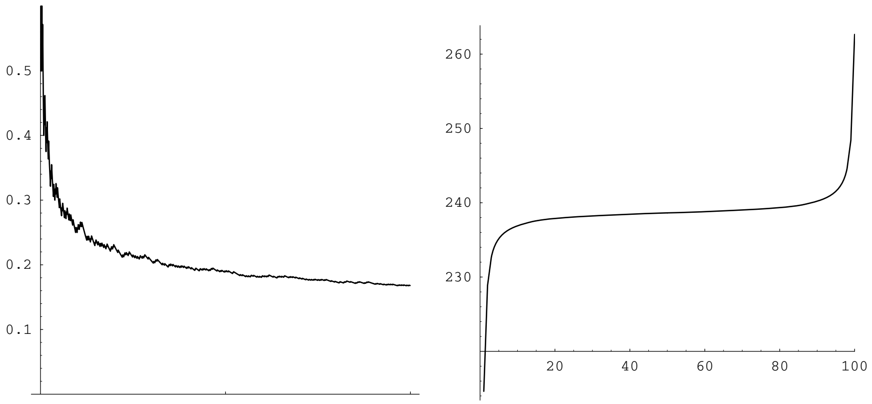


Fig. 7. Walk on the indicator $v(k)$ (left) and its power spectrum

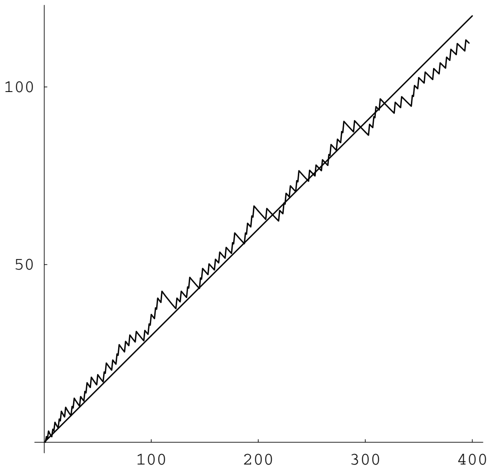


Fig. 8. Complexity for the first 400 primes and its corresponding least square fit

to notice that although there is an increasing complexity, for the first numbers of the sequence $n \leq 400$ there is a constant trend to complexity $\alpha \cong 0.3$ which is given by the least square fit (Fig. 8).

5 Conclusion

In this paper the existence of correlation in prime distribution has been discussed. Some patterns were shown by using the indicator matrix and the fractal dimension has been given.

References

1. Ares, S., Castro, M.: Hidden structure in the randomness of the prime number sequence? Physica A 360, 285–296 (2006)
2. Arneado, A., Bacry, E., Graves, P.V., Muzy, J.F.: Characterizing long-range correlations in DNA sequences from wavelet analysis. Phys. Rev. Lett. 74, 3293–3296 (1995)
3. Arneado, A., D'Aubenton-Carafa, Y., Audit, B., Bacry, E., Muzy, J.F., Thermes, C.: What can we learn with wavelets about DNA sequences? Physica A 249, 439–448 (1998)
4. Berger, J.A., Mitra, S.K., Carli, M., Neri, A.: Visualization and analysis of DNA sequences using DNA walks. Journal of The Franklin Institutes 341, 37–53 (2004)
5. Bernaola-Galván, P., Román-Roldán, R., Oliver, J.L.: Compositional segmentation and long-range fractal correlations in DNA sequences. Phys. Rev. E 55(5), 5181–5189 (1996)
6. Cattani, C.: Wavelet Algorithms for DNA Analysis. In: Elloumi, M., Zomaya, A.Y. (eds.) To appear on Algorithms in Computational Molecular Biology: Techniques, Approaches and Applications. Wiley Series in Bioinformatics, ch. 35. Wiley-Blackwell, Chichester (2010)
7. Coward, E.: Equivalence of two Fourier methods for biological sequences. Journal of Mathematical Biology 36, 64–70 (1997)
8. Dodin, G., Vandergheynst, P., Levoir, P., Cordier, C., Marcourt, L.: Fourier and Wavelet Transform Analysis, a Tool for Visualizing Regular Patterns in DNA Sequences. J. Theor. Biol. 206, 323–326 (2000)
9. Edwards, H.M.: Riemann's zeta-function. Academic Press, London (1974)
10. Herzel, H., Trifonov, E.N., Weiss, O., Grosse, I.: Interpreting correlations in biosequences. Physica A 249, 449–459 (1998)
11. Li, W., Kaneko, K.: Long-range correlations and partial $1/f^{\alpha}$ spectrum in a noncoding DNA sequence. Europhys. Lett. 17, 655–660 (1992)
12. Littlewood, J.E.: Sur la distribution des nombres premieres. C. R. Acad. Sci. Paris 158, 1869–1872 (1914)
13. Narkiewicz, W.: The development of prime number theory. Springer, Heidelberg (2000)
14. Peng, C.-K., Buldryev, S.V., Goldberg, A.L., Havlin, S., Sciortino, F., Simons, M., Stanley, H.E.: Long-range correlations in nucleotide sequences. Nature 356, 168–170 (1992)
15. Peng, C.-K., Buldryev, S.V., Havlin, S., Simons, M., Stanley, H.E., Goldberg, A.L.: Mosaic organization of DNA nucleotides. Phys. Rev. E 49, 1685–1689 (1994)
16. Shapiro, H.H.: Introduction to the Theory of Numbers. John Wiley & Sons, New York (1983)
17. Schlesinger, M.: On the Riemann hypothesis: a fractal random walk approach. Physica A 138, 310–319 (1986)
18. Voss, R.F.: Evolution of Long-Range Fractal Correlations and $1/f$ Noise in DNA Base Sequences. Physical Review Letters 68(25), 3805–3808 (1992)
19. Weiss, O., Herzel, H.: Correlations in protein sequences and property codes. J. Theor. Biol. 190, 341–353 (1998)

Efficient Energy Supply from Ground Coupled Heat Transfer Source

Maurizio Carlini[1] and Sonia Castellucci[2]

[1] DiSAFRi, Department forest environment and resources,
University of Tuscia, Via San Camillo de Lellis s.n.c, 01100 Viterbo, Italy
maurizio.carlini@unitus.it
[2] SEA Tuscia s.r.l. Spin-off University of Tuscia, Via del Suffragio n.1, 01100 Viterbo, Italy
sonia.castellucci@libero.it

Abstract. The increasing demands of Energy for industrial production and urban facilities, asks for new strategies for Energy sources. In recent years an important problem is to have some energy storage, energy production and energy consumption which fulfill some environment friendly expectations. Much more attention has been recently devoted to renewable energies [1]. Among them energy production from geothermal sources has becoming one of the most attracting topics for Engineering applications. Ground coupled heat transfer might give an efficient energy supplies for well-built construction. At a few meters below the earth's surface the underground maintains a constant temperature in a approximation through the year allowing to withdraw heat in winter to warm up the habitat and to surrender heat during summer to refresh it. Exploiting this principle, heat exchange is carried out with heat pumps coupled with vertical ground heat exchanger tubes that allows the heating and refreshing of the buildings utilising a single plant installation. This procedure ensure a high degree of productivity, with a moderate electric power requirement compared to performances. In geographical area characterize by specific geological conformations such as the Viterbo area which comprehend active volcanic basins, it is difficult to use conventional geothermal plants. In fact the area presents at shallow depths thermal falde ground water with temperatures that varies from 40 to 60°C geothermal heat pumps cannot be utilized [2]. In these area the thermal aquifer can be exploited directly as hot source using vertical heat exchanger steel tubes without altering the natural balance of the basin. Through the heat exchange that occurs between the water in the wells and the fluid that circulates inside the heat exchanger, you can take the heat necessary to meet the needs. The target of the project is to analyze in detail the plant for the exchange of heat with the thermal basin, defining the technical-scientific elements and verifying the exploitation of heat in the building-trade for housing and agricultural fields.

Keywords: heat, thermal aquifer, thermal energy.

1 Introduction

The geology of the volcanic basin area in the Italian region of Viterbo, defines a situation which is very rich and unique, from the point of view of energy source, but it

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 177–190, 2010.

is difficult to be approached with the conventional low-enthalpy geothermal systems for domestic, industrial and agricultural applications.

Ground water springs, are a constant feedback with limited depth and a temperature range from 40 to 60 degrees centigrade[3].

Therefore, the respective locations of these volcanic basins, do not allow for usual exploiting geothermal technology, but require a series of experiments and devices aimed at the following tasks:

1. heat transfer with vertical closed-loop equipments
2. avoid the depletion of the source or aquifer
3. operate with highly environmentally friendly technologies
4. avoid drawing groundwater, so that the balance of drawing is only the heat exchange
5. develop technologies able to harness the gaseous fluids in groundwater and not allow the release of the same
6. develop technologies aiming at low cost installation and operation
7. develop technologies which can be suitable also for small systems or livestock farm or isolated residential unit
8. Developing technologies in synergy with photo-voltaic production and/or wind, solar thermal productions.

The economic structure of the Viterbo province, mirrors the national trend in developing localized companies with small and medium size.

Specifically, this province is characterized, in agriculture, by a high concentration of poultry and sheep farms whose maximum demand is their need to produce and store thermal energy for heating during winter months[4].

The widespread presence on the territory of thermal aquifers represents a possible optimal solution compatible with this kind of economics.

The main task of this research is to give a feasible answer to the specific requests of rural companies in the Viterbo province, and to develop well coded solutions suitable for the whole national territory.

This research aims to consolidate the technical-scientific knowledge for the exploitation of the aquifers, even very deep, by using vertical probes respectful of the aquifer, and operating only the heat exchange without dispersion of thermal water [5]. The purpose of this study is to assess, from a theoretical point of view, the rate of heat transfer undertaken in the experimental plant. These computations will make it possible to develop the optimal heat-transfer technology which could be used for allowing a minimal intervention both in relatively small settlements such as villas or apartments and in large plants demanding higher volumes such as residential buildings or animal farms and greenhouses.

The system to be assessed during this initial phase, to be implemented during the experimental tests, is made by “U”-like steel probes inside a thermal pit, designed in a such way to ensure heat exchange by contact without interfering on the thermal aquifer energy balance[6].

In particular, this work intends to compute the theoretical amount of heat which can be drawn from the geothermal wells, and also intends to present the engineering components which are needed for the realization of experimental trials.

1.1 Regulations and Laws on the Production of Low Enthalpy Geothermal Energy

The electricity obtained from geothermal energy source was produced in Italy for the first time in 1904 in the Tuscany town of Lardarello, which is located on the same geological hydrothermal area of Viterbo (see Fig. 4)[7]. At that time, Italian regions were ruled by different laws resulting from the application of the existing regulations inherited by the Italian unification. In many regions, including Tuscany, which owns more than 90% of Italian geothermal resources, outweighed the interests of landowners. This means that they possessed the land and nothing could be extracted from underground [8]. However, the division of land among different, often fighting, landholders has hindered the development of mining and geothermal energy exploiting since the landowners even did not possess the technical knowledge and financial resources.

That is why in 1927 the Italian government, passing by the abolition of existing laws and the constitution of a single national law, thought it was necessary to regulate mining together with the exploitation of geothermal energy, so that the responsibility is enrolled by the mining activity[9].

Thanks to the new legislation "*the rights of landowners must be closely linked to the needs of the community*" so that "*the exploitation of the subsoil is independent on the surface profits*" and, as a consequence, the title of the property itself.

Thanks to this law the Ministry of National Economy, was solely responsible for relations between Researchers and / or landowners in respect of the development of coal mining.

Currently, the Italian legislation on the exploitation of geothermal resources is governed by the following laws:

- Regio Decreto 29 luglio 1927, n. 1443: "Legislative rules to govern the exploration and production of mines in the Kingdom" posted in Gazz. Uff. 23 agosto 1927, n. 194.
- DPR 9 aprile 1959, n. 128: Police regulations about mines and quarries, published in Gazz. Uff. 11 april 1959, n. 87,S.O.
- In 1986, a new law was approved aiming to accommodate these new needs arising, in particular, from the method of exploitation of geothermal resources. The main concerns of this law can be found in: *Art 2. Inventory of geothermal resources". Art. 4. Research permits and provisions to safeguard integrity both of the environment and urban settlements. Art. 5. Extent and duration of the exploration license. Art. 11. Provisions to safeguard the integrity the environment, to keep the eco system ecological balance and the integrity of city planning.*
- Constitutional Law, October 18, 2001, n. 3 titled "Changes to Title V of Part II of the Constitution" art. 3, This law establishes the matters of exclusive legislation of the State, as regards the energy sector which is the sole responsibility of the State and give rules for "*transport and distribution of energy";*

Some Interesting rules for the low enthalpy geothermal energy are contained in the law **L. 99/2009** entitled “Provisions for the development and internationalization of enterprises, and energy”, where the deployment of low enthalpy geothermal plants are given.

The use of low enthalpy geothermal plants is included in all legislation concerning the rational use of energy which were ratified in recent years.

In the wake of the regional legislative autonomy all Italian regions have a role in regulating their own exploitation of geothermal sources. As an example the Region Lombardia, with law n. 1 of 05/01/2000 identifies those functions transferred or delegated to local authorities with the autonomous activities and those kept in the head region. Tuscany Region also specified the competence and the authorizations necessary for the construction of geothermal plants (see following table).

Type of Plant	**Authorization**	**Procedure**
Depth below < 400 m Thermal Power < 2.000 Kw t Without fluid withdrawal	Regional (Artt. 11 e 15 L.R.39/05)	Application to Regional Administration through a specific form required to the region
Depth below < 400 m Thermal Power < 2.000 Kw t Without fluid withdrawal	Regional (Artt. 11 e 15 L.R.39/05)	Application to Regional Administration. Region will activate a unique procedure concerning all duties.

2 The Basic Model of Heath Exchange with Natural Convection

The thermal energy used is transferred by natural convection from the thermal water to the heat exchanger in the well. The heat exchanger consists of a U-shaped tube where inside there is, as circulating fluid, water [10].

The testing system is simulated by a tank containing hot water at rest in which is inserted a U-shaped tube exchanger.

Because of the geology of the area identified for the trial, it is assumed that the water temperature of the well is constant [11].

The starting point is the general equation for heat transfer across a surface which is:

$$Q = A \cdot U_D \cdot \Delta T_{LM} \tag{1}$$

Q : heat transferred per unit time, W
A : heat-transfer area, m^2
U_D : overall heat transfer coefficient, W/m^2 ^{o}C
ΔT_{LM} : mean temperature difference, the temperature driving force, ^{o}C :

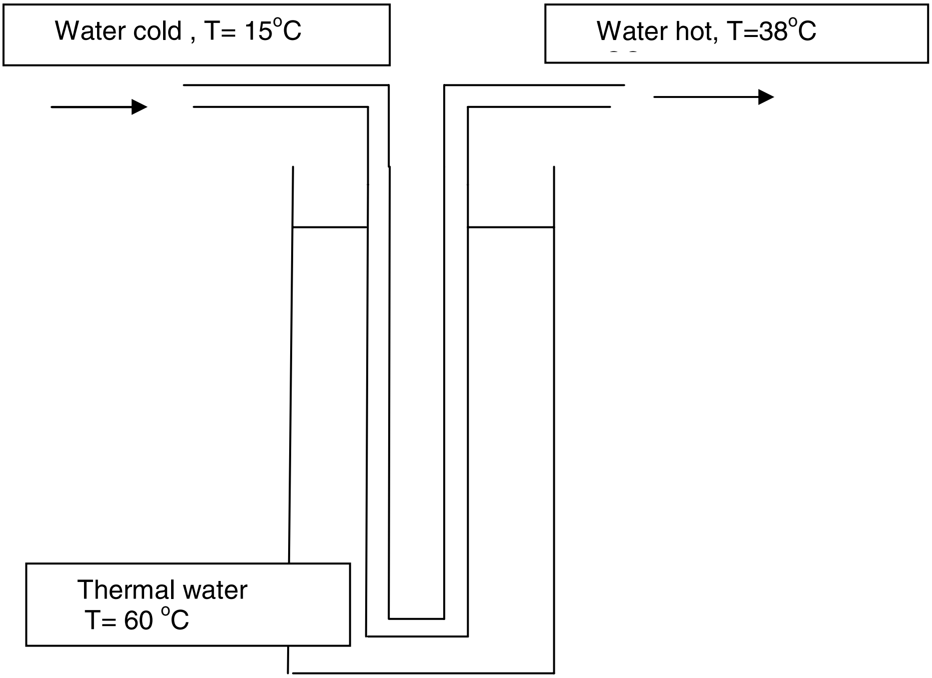


Fig. 1. Schematic heat exchanger in the well

$$\Delta T_{LM} = \frac{(T-t_2)-(T-t_1)}{\ln\left[\frac{(T-t_2)}{(T-t_1)}\right]} \tag{2}$$

T : temperature hot fluid
t_1 : temperature cold fluid, in the feeding pipe
t_2 : temperature cold fluid, in the outgoing pipe

For the heat exchange across a typical heat-exchanger pipe the relationship between the overall coefficient and the individual coefficients, which are the reciprocals of the individual resistance, is given by [12]:

$$\frac{1}{U_D} = \frac{1}{h_{i0}} + \frac{d_0}{2k}\ln\frac{d_i}{d_0} + \frac{1}{h_0} + R \tag{3}$$

U_D : the overall coefficient based on the outside area of the tube, W/m^2 °C
k : thermal conductivity of the tube wall material, steel 50 W/m °C
d_i : inner pipe diameter, m
d_0 : outer pipe diameter, m

h_{i0} : outside dirt coefficient (fouling factor), W/m^2 °C
h_0 : outside fluid film coefficient, W/m^2 °C

R : dirt factor pipe for water 0,0002 (m^2h °C/kcal) for T < 50°C and 0,0004 (m^2h °C/kcal) for T>50°C

The temperature reached by the heat-carrying fluid outgoing from the pipe should be at least about 38° C, the minimum value needed to activate the low temperature heating systems. The value of heat needed to warm up the water is given by:

$$Q = c_p \, w \, (t_2 - t_1) \tag{4}$$

c_p : specific heats water, (J/kg °C)
w : flow water, kg/s.

From this equation it is possible to compute the heat available from the heat exchanger, at a fixed geometry, being the size of the well given.

2.1 Heat Transfer Coefficient

The first step is to calculate the overall heat transfer coefficient according to the following scheme:

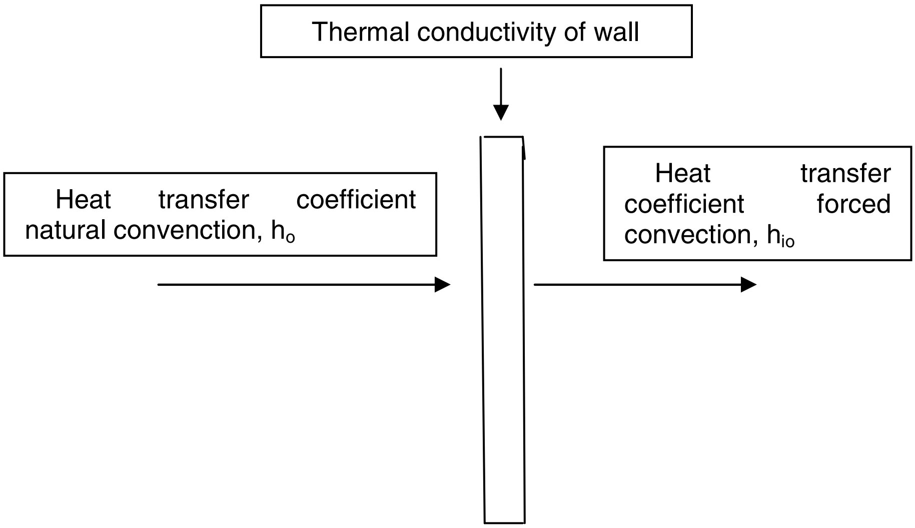

Thermal conductivity of wall
Heat transfer coefficient natural convenction, h_o
Heat transfer coefficient forced convection, h_{io}

Fig. 2. Heat transfer coefficients

2.1.1 Forced Convection

Inside the pipe the heat transfer is due to a forced convection, since the fluid is pumped into the tube.

The heat transfer coefficient is given by the equation:

$$h_{i0}(\mathrm{Re}) = \frac{1}{d_{i0}} k^{2/3} c_p^{\;1/3} \mu^{1/3} \left(\frac{\mu}{\mu_0}\right)^{0.14} J_h(\mathrm{Re}) \tag{5}$$

d_{i0} : outer diameter, m
k : thermal conductivity for water, 0.68 W/mK
c_P : specific heats water, 4196 J/kg °C
μ : viscosity for water, $0.38 \cdot 10^{-3}$Pa·s
J_h : heat transfer coefficient, J_H=f(Re)
Re : Reynolds number

Reynold number is defined as:

$$\mathrm{Re}=(v d_i \rho)/\mu \tag{6}$$

being
v : velocity of the heat-carrying fluid outgoing from the pipe
d_i : inner pipe diameter, m
ρ : density
μ : viscosity.

The fluid motion in a duct is considered laminar if the numerical value of Re is less than 2100, turbulent if more than 10000. Transition regime is when 2100 < Re <10000. The heat transfer coefficient J_H might assume different values depending on the type of motion of the fluid inside the tubes:

$$J_h(\mathrm{Re})=\begin{cases} 1.86\left(\mathrm{Re}\dfrac{d_i}{L}\right)^{1/3} & , \quad \mathrm{Re}<2100 \\ 36.45-\left[36.45-1.86\left(\mathrm{Re}\dfrac{d_i}{L}\right)^{1/3}\right]\left(1.2658-1.2658\,10^{-4}\,\mathrm{Re}\right) & , \quad 2100<\mathrm{Re}<10000 \\ 0.023\,\mathrm{Re}^{0.8} & , \quad 10000<\mathrm{Re} \end{cases} \tag{7}$$

2.1.2 Natural Convection

Inside the shaft there is thermal water which is not forced to move by any external action, the only present are convective motions which are consequence of local variations in density. Convective motions are the only responsible for heat transfer. The coefficient of transmission outside a horizontal tube immersed in a liquid at rest can be computed by the equation:

$$h_0 = 0.51\frac{k}{d_0}\left\{\frac{\left[d_0^{\,3}\rho^2\, g\,\beta\left(T_0-T_p\right)\right]c_p}{\mu\, k}\right\}^{1/4} \tag{8}$$

k : thermal conductivity
d_0 : outer pipe diameter, m

ρ : density
g : gravity constant, 9.81 m/s^2
β : coefficient of expansion
T_p : temperature outside the wall pipe
T_0 : temperature water in the well
c_P : specific heat of the water
μ : viscosity coefficient

All parameters have to be evaluated at the average temperature of the film T_f which is established between the tube wall and fluid at rest.

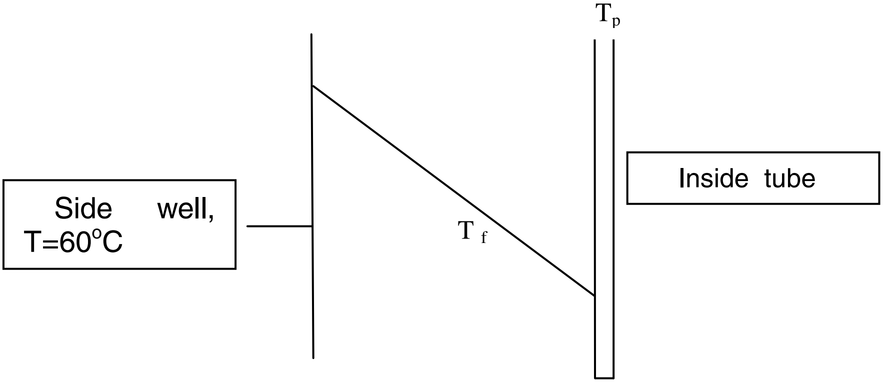


Fig. 3. Scheme heat exchange in the film

The average temperature T_f of the film can be obtained by successive approximations from the equation

$$h_{i0}\left(T_i - T_p\right) = h_0\left(T_0 - T_p\right) \tag{9}$$

T_i : mean temperature of the fluid inside the pipe
T_p : temperature wall
T_0 : mean temperature of the water in the well,

with

$$T_p = \frac{h_{i0}T_i + h_0T_0}{h_{i0} + h_0} \tag{10}$$

$$\overline{T}_f = \frac{T_p + T_0}{2} \tag{11}$$

T_f : mean temperature of the film

$$\beta = -\frac{1}{\rho}\frac{d\rho}{dT} \tag{12}$$

ρ: density, $\rho = f(T)$

$$\rho = 1.00504 - 0.000422027\,T \tag{13}$$

2.2 Computation of the Heat Exchange

For the theoretical calculation of the heat, which can be taken from the well, the spring was considered at a constant high temperature (hot water). This assumption is supported by the presence in the area of high volcanic activity.

Input values:

w = 0.1 kg/s
c_p= 4196 J/kg °C
t_1= 15 °C
t_2= 38 °C
T= 60 °C
P= 1000 kg/m^3
μ= 0.38 · 10^{-3} Pa·s

According to (4), the heat Q which is required to warm up the fluid (water) from the temperature T_1 to temperature T_2 is,

$$Q = 9650.8 \text{ W} \tag{14}$$

The amount of heat, taken from a fixed geometry given by the U-shaped tube inserted into the well, might be computed from Eq. (1).

For the calculation of A is necessary to know the size of the U-tube:

d_{io} = 33.7 mm
d_i = 29.1 mm
L = 110 m, pipe length

$$A = 23.21 \tag{15}$$

The logarithmic mean temperature difference is:

$$\Delta T_{ML} = 24.2 \tag{16}$$

Thus we have to proceed with the computation of the overall heat transfer coefficient, according to Eq. (3). In particular, the coefficient of heat transfer fluid inside the pipe depends on the Reynolds number [13]:

$$\mathrm{Re}= 11563 \quad . \tag{17}$$

In presence of phase then the coefficient of heat transfer and (5) give:

$$h_{io}= 1096.72\ \mathrm{W/m^2K} \tag{18}$$

Regarding the coefficient of heat transfer fluid outside the tube which is in natural convection, it can be obtained from (8) by successive approximations.

The coefficient of thermal expansion and density, have to be computed at the temperature of the film:

$$\beta = \frac{0.000422027}{1.00504 - 0.000422027\,T} \tag{19}$$

By successive approximations we have obtained the following values for h_o:

h_o	T_P	T_f	ρ	β	$h_{(o)}$
800	40.66	50.33	0.9837	$4.28 \cdot 10^{-4}$	27.218
27.218	32.57	46.285	0.9855	$4.28 \cdot 10^{-4}$	29.71
29.71	33.02	46.51	0.9854	$4.28 \cdot 10^{-4}$	29.59
29,59	33.002	46.501	0.9854	$4.28 \cdot 10^{-4}$	29.59

The global coefficient of heat transfer, considering all factors of (3) [14]:

$$U_D = 28.63\ \mathrm{W/m^2\ {}^{o}C} \tag{20}$$

Heat Q can be detected from the well through the heat exchanger as given by (1):

$$Q = 16080.95\ \mathrm{W} \tag{21}$$

According to the value calculated in (14) we have assumed both that the heat exchange system is sufficient to warm up the water for the low temperature heating system and there is a very low total pressure lost in the pipes:

$$\Delta P= 1.69 \cdot 10^{-2}\ \mathrm{bar}$$

3 Models and Methods

We propose the realization of an experiment to assess the amount of heat that can be taken from a thermal spring in which hot water is present. Heat is taken without draining-off the water.

The trial involves the carrying out of several steps needed to proceed in a rigorous evaluation of the heat taken from the well. In particular, once opened, the shaft will be inserted by instrumentation for monitoring water temperature and the heat exchanger performances. The whole plant will be equipped with instrumentation able to control temperature and flow rate which are necessary to known for the calculation of heat that can be taken from the well [15].

It is also planned a preliminary measure of thermal activities of the neighboring wells in order to have a clear idea of the temperatures involved and about the possibility of identifying potential changes related to testing under consideration.

The experimental program will be implemented as follows:

- Opening and securing the well: Once the well has been opened, the water at the surface mouth well will be lowered and the water loss repaired
- Insertion of monitoring equipment in the well before the activation of the pilot plant it is planned a series of measurements concerning the characteristics of the well needed to measure the temperature gradient and the organoleptic characteristics of the thermal water. It is also expected to monitor the temperature of thermal basins adjacent to the shaft. By using these data it is possible to test the impact on environment, concerning the heat draw from the well.
- Construction of the geothermal plant: it involves the insertion of 4 probes that will be prepared on site, through the implementation of the joints necessary for assembly. Special pieces will be realized in loci. The probes are designed in a such way that they can be used either in series or in parallel in this way one can check the best link that provides the highest heat exchange. Controls are provided both for temperature and flow on the outlet and the return of the probes inserted necessary to verify the heat. The fluid used inside the probe is water which is taken from the cold sink. It has been also provided a device enabling the heat flow in order to make it possible to measure the heat exchange.
- Simulation by the software TRNSYS of the conditions under which the heat exchange is realized. Based on data collected during the test, some simulations will be carried out on computer to ensure the maximum efficiency of the plant.

3.1 Description of the Site and the Well

The well, interesting to experiment, is located at the hatchery at the center of the basin in the City of Viterbo (Fig. 4). As one can see from Fig. 4 Viterbo is located in one of the most active geothermal Italian areas.

Well looks like an artesian well of 150mm in diameter and a depth of about 60m, at the wellhead is located a steel tube bent at the summit needed to hold the swing of thermal water up to 1.5 m from the surface. In the vicinity of the well there is both the water supply to be used as heating fluid and the electric current needed to feed all utilities for running the plant activity [16].

3.2 Description of Measures and Controls

To conduct the experiment it is necessary to examine in detail the internal temperature of the well, and the characteristics both input and output of fluid in the plant. For this reason, it is envisaged the use of three temperature probes to be inserted at regular

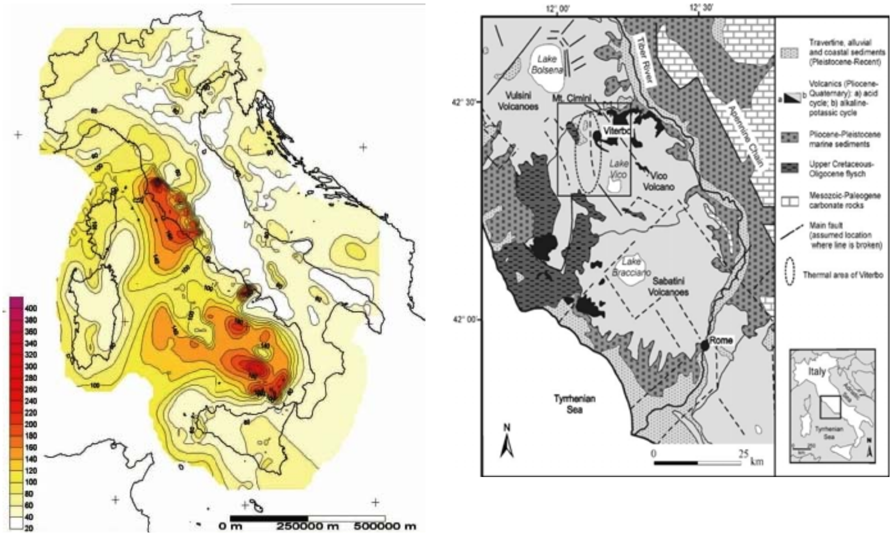


Fig. 4. Viterbo hydrothermal area within the regional geological picture[16]

intervals within the well, so that one can see and measure the temperature inside the well. There will be, at constant time intervals, some sampling to evaluate pH, electrical conductivity, density and salinity. These parameters are needed to evaluate the possible corrosion of the materials included in the well.

All sensors involved in the experiment need a framework for acquiring and monitoring data [17].

The framework consists of a fiberglass container with two compartments, one containing the industrial PC and the other a PLC that contains the electronics interface.

From the main monitor some cables connect the data acquisition devices. This configuration it is also ready for future implementation of automation systems and control valves, for the use of heat exchangers in series and parallel, and the choking of the flow into exchanger according to parameters set.

Furthermore, this system provides for controls of the temperature and flow on the outlet and the return of the heat exchanger [18]. These measures are necessary to calculate the heat exchanged in the well, thus enabling us to verify the temperature reached by the fluid circulating in the exchanger.

3.3 Plant Description

The heat exchanger to be undertaken consists of vertical U-shaped steel probes that may have different geometries [19]:

- a single U-shaped pipe: inside the same drill it is placed both the ongoing tube and the outgoing tube which are connected at the bottom.

- a double U-shaped pipe: is realized as the previous one, except that in the drill there are four tubes pair wise connected at the bottom.
- coaxial pipe: the outgoing tube is inside the ingoing tube. The diameter of the ingoing tube nearly coincide with the diameter of the drill.
- coaxial pipe with complex geometry: similar to the previous one with the only difference that between the inside and outside tubes there are some connecting wings enabling a better heat exchange. The returning fluid instead of the inner pipe can circulate in some of the peripheral channels so that it can exchange heat with the ground in both directions of its circular path [20].

In the first analysis was assessed the amount of heat taken from the U-shaped pipe inserted into the shaft, the tube used has the outer diameter of 33.7 mm and the inner diameter of 29.1 mm, it provides a length of heat exchange along 110 m.

4 Conclusion

Based on the calculations we have done, we have shown that, from a theoretical point of view, the designed heat exchange, in the given conditions, is sufficient to warm up water for the realization of a low temperature heating system. We have made several assumptions in the evaluation of heat transfer from the system, which however must be verified during the experimental trials. The first parameter to be checked is the water temperature of the well, in fact, in the design it is assumed to be constant along the entire length of the heat exchanger. We can predict that surely there will be a temperature gradient as a function of the depth of the well. It should be also verified the variation in time of the heat exchange temperatures.

References

1. Lazzarin, R.: Ground as a possible heat pump source. Geothermische Energie 32/33, marzo/giugno (2001), http://www.geothermie.de/gte/gte3233/ground_as_a_possible_heat_pump_s.htm
2. Ingersoll, L.R., Zobel, O.J., Ingersoll, A.C.: Heat conduction: with engineering and geological applications, 2nd edn. (1954)
3. Rybach, L., Scanner, B.: Ground-source heat pump system - The European Experience. In: GHC Bulletin, marzo 2000 (2000), http://geoheat.oit.edu/bulletin/bull21-1/art4.pdf
4. Driver, A.W.: Groundwater as a heat source for geothermal heat pumps. In: International Geothermal Days, Germany (2001)
5. Talleri, M.: Applicazioni geotermiche negli impianti di attivazione termica della massa, Seminari Velta 2001 (2001)
6. Kavanaugh, S.P., Rafferty, K.: Ground source heat pumps - Design of geothermal systems for commercial and institutional buildings. A.S.H.R.A.E. Applications Handbook (1997)

7. Bonacina, C., Cavallini, A., Mattarolo, L.: Trasmissione del calore. Ed. Cleup, febbraio (1994)
8. Olesen, B.W., Meierhans, R.: Attivazione termica della massa. Seminari Velta, (2001)
9. Yavuzturk, C.: Modelling of Vertical Ground Loop Heat Exchangers for Ground Source Heat Pump Systems. Thesis to the Faculty of the Graduate College of the Oklahoma State University in partial fulfilment of the requirements for the Degree of Doctor of Philosophy (December 1999)
10. Eskilson, P.: Thermal Analysis of Heat Extraction Boreholes. Doctoral Thesis, University of Lund, Department of Physics, Sweden (1987)
11. Hellström, G., Sanner, B.: PC-programs and modelling for borehole heat exchanger design. In: International Summer School on Direct Application of Geothermal Energy,
12. Sebastiani, E.: Lezioni di Impianti chimici, edizioni scientifiche SIDERA
13. Bakhoum, E., Toma, C.: Mathematical Transform of Travelling-Wave Equations and Phase Aspects of Quantum Interaction. Mathematical Problems in Engineering 2010, Article ID 695208,(2010), doi:10.1155/2010/695208
14. Toma, G.: Specific Differential Equations for Generating Pulse Sequences. Mathematical Problems in Engineering 2010 Article ID 324818, (2010), doi:10.1155/2010/324818
15. Thornton, J.W., McDowell, T.P., Shonder, J.A., Hughes, P.J., Phaud, D., Hellstrom, G.: Residential Vertical Geothermal Heat Pump System Models: Calibration to Data. ASHRAE Transactions 103(2), 660–674 (1997)
16. Piscopo, V., Barbieri, M., Monetti, V., Pagano, G., Pistoni, S., Ruggi, E., Stanzione, D.: Hydrogeology of thermal waters in Viterbo area, central Italy. Hydrogeology Journal 14, 1508–1521 (2006)
17. Mei, V.C., Emerson, C.J.: New Approach for Analysis of Ground-Coil Design for Applied Heat Pump Systems. ASHRAE Transactions 91(2), 1216–1224 (1985)
18. Muraya, N.K.: Numerical Modelling of the transient thermal interference of vertical U-tube heat exchangers, Ph. D. Thesis, Texas A&M University, College Station (1995)
19. Rottmayer, S.P., Beckman, W.A., Mitchell, J.W.: Simulation of a Single Vertical U-tube Ground Heat Exchanger in a Infinite Medium. ASHRAE Transactions 103(2), 651–659 (1997)
20. Shonder, J.A., Beck, J.V.: A New Method to Determine the Thermal Properties of Soil Formations from. In: Situ Field Tests.In: OAK RIDGE NATIONAL Laboratory, managed by Ut-Bettelle for the Department of Energy (April 2000)

A Study of Nonlinear Time–Varying Spectral Analysis Based on HHT, MODWPT and Multitaper Time–Frequency Reassignment

Pei-Wei Shan and Ming Li

School of Information Science & Technology, East Normal University,
No. 500, Dong–Chuan Road, Shanghai 200241, China
midoban43@yahoo.com.cn, ming_lihk@yahoo.com

Abstract. Numerous approaches have been explored to improve the performance of time–frequency analysis and to provide a sufficiently clear time–frequency representation. Among them, three methods such as the empirical mode decomposition (EMD) with Hilbert transform (HT) (or termed as the Hilbert–Huang Transform (HHT)), along with the Hilbert spectrum based on maximal overlap discrete wavelet package transform (MODWPT) and the multitaper time–frequency reassignment raised by Xiao and Flandrin, are noteworthy. This study evaluates the performances of three transforms mentioned above, in estimating single and multicomponent chip signals in the presence of noise or noise–free. Rényi Enropy is implemented for measuring the effectiveness of each algorithm. The paper demonstrates that under these conditions MODWPT owes better time–frequency resolution and statistical stability than the HHT. The multitaper time–frequency reassigned spectrogram makes excellent trade–off between time–frequency localization and local stationarity.

Keywords: Hilbert–Huang transform, MODWPT, Multitaper, Time–frequency reassignment, Rényi entropy.

1 Introduction

Time–frequency (TF) or time–scale representations are widely used for nonstationary signal analysis. Among these, the spectrogram, although probably being one of the earliest and still one of the commonly used, has severe shortcomings, since there exist, not just theoretically, difficulties in accurately estimating the signal instantaneous frequency and group delay, but also practically, a trade–off between time and frequency resolutions [1][2]. To overcome these significant drawbacks, several methods have been raised to improve the performance of TF analysis and to provide a sufficiently clear time–frequency representations. Among them, three approaches have gained increased attention, the wavelet analysis, empirical mode decomposition (EMD) with Hilbert transform (HT) (Huang et al. 1998) and the multitaper TF reassignment raised by Xiao and Flandrin in 2007.

In a number of studies, EMD + HT (or termed as the Hilbert–Huang Transform (HHT)) has been advocated by illustrating its unique self–adaptive properties [3].

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 191–205, 2010.

However, several problems still exist in EMD and associated Hilbert transform. Such as, the method yet lacks a complete theoretical basis and in particular the EMD method produces oscillatory or poorly defined Hilbert spectra, with evidence of notable mode mixing, end effects and so on [4][5]. Although so far, all these are pending problems to solve, EMD + HT owes strength of being data dependent and provides a potentially viable method. Furthermore, it offers a new sight for non–stationary signal processing, that is, individual component signal with physically meaningful instantaneous frequency can be obtained by appropriate signal decomposition method.

To avoid the deficiencies in HHT, S. Olhede and A.T. Walden have developed a new wavelet–based algorithm, namely, maximal–overlap discrete wavelet packet transform (MODWPT) [6]. The ordinary discrete wavelet transform (DWT) requires the sample size to be exactly a power of 2 for the full transform. Besides, the wavelet and scaling coefficient of DWT are not circularly shift equivariant. By avoiding down sampling, MODWPT overcomes these disadvantages of DWT. With optimum decomposition scale and disjoint dyadic decomposition, the complicated signal could be decomposed into a number of single component signals with instantaneous frequency physically meaningful. Furthermore, each single component signals obtained by MODWPT has desirable statistical characteristic, which are desirable properties to deal with non–stationary time series in practice [7].

Several methods for nonstationary signal representations have been proposed among the Cohen's class of bilinear TF distributions to replace the "classical" Fourier based spectrogram and Wigner–based distributions. Nevertheless, there commonly exists a critical problem of these methods, their readability, which means both a good concentration of the signal components and no miss leading interference terms [8]. For this reason, a dramatic improvement has been made by means of the reassignment technique, which provides a sharp visual TF distribution. In addition, for an appropriate readability, a representation must either be self–adaptive or using automatic procedures [8]. Thus it comes to us the idea of multitapering, pioneered by D.J. Thomson in a stationary setting [9]. It improved the statistical stability without having recourse to a time–averaging step, and offered a low variance spectrum without degrading the resolution of line components. However, the method exists two problems: 1) The windows used by Thomson are not optimal in a time–frequency setting, 2) The chirping rate of the line components must be very small so that they can be approximated as piece–wise sinusoids [10]. To refine the Thomson's method into an improved spectrum estimator, numerous attempts of extending multitaper techniques to nonstationary situations have been made, and Xiao and Flandrin raised a method that consists in combing TF reassignment with multitapering [11], which we called Xiao–Flandrin algorithm. The principle of multitaper is based on a family of orthogonal functions: Hermite functions (HF) and the essence of the reassignment technique consists in evaluating for each TF location (the center of mass) and reassigning the spectrogram value to this location. Combining two techniques above makes efficient improvement on the TF localization trade–off and provides us a relatively good analysis method which offers better local stationarity and smoothness and also needs no priori knowledge compared to the former method: MODWPT.

The organization of this paper is given as follows. The rationale of three TF methods will be elaborated separately in Section 2 ~ 4. Performances of each method in representing time–frequency characteristic will be compared and respective efficiency measuring of the three representations by means of Rényi entropy will also be shown in Section 4. Finally, we will offer the conclusion in Section 5.

2 Brief Introduction of HHT

One way to express the nonstationarity is to find instantaneous frequency and instantaneous amplitude. Although the definition of instantaneous frequency is controversial, it is well founded to describe that for a given length of signal, there is only one frequency value within the length of the signal or the signal is monocomponent [12]. Therefore, Huang presented a method called EMD to decompose any multicomponent signal into a set of nearly monocomponent signals termed intrinsic mode functions (IMFs).

Empirical Mode Decomposition

Physically speaking, the necessary conditions to define a meaningful instantaneous frequency are that the signal must be symmetric concerning the local zero mean, and have the same numbers of zero crossings and extrema. This means, in an IMF function, the number of extrema and the number of zero crossings must be either equal or different at most by one in the whole data set, and the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero at every point. All these conditions are so strict that the determined IMF may not satisfy them precisely. Consequently, the resultant IMF is nearly a monocomponent function, while not a perfect one.

The EMD is developed based on the assumption that any signal consists of many different IMFs. The procedures to decompose signal $x(t)$ can be enumerated as following steps:

a) Find all the local maxima from $x(t)$ and connect them with the cubic spline to form the upper envelope denoted by $x_{\mathrm{up}}(t)$.
b) Find all the local minima from $x(t)$ and connect them with the cubic spline to form the lower envelope denoted by $x_{\mathrm{low}}(t)$.
c) Let $m_{11}(t) = [x_{\mathrm{up}}(t) + x_{\mathrm{low}}(t)]/2$.
d) Subtract $m(t)$ from the signal: $p_{\mathrm{i}}(t) = x(t) - m_{\mathrm{i}}(t)$.
e) Return to step (a) and stop when $x(t)$ remains nearly unchanged.
f) Once we obtain an IMF, $c_{\mathrm{i}}(t)$, remove it from the signal $p_{\mathrm{i+1}}(t) := p_{\mathrm{i}}(t) - m_{\mathrm{i}}(t)$ and return to (a) if $x(t)$ has more than one extremum (neither a constant nor a trend), called as the residue $r(t)$.

By using EMD, the signal $x(t)$ can be expressed as the combination of IMF $c_{\mathrm{i}}(t)$ and the residue $r_{\mathrm{n}}(t)$. Expressed as follow [12]

$$x(t) = \sum_{i=1}^{n} c_i(t) + r_n(t) \tag{1}$$

Then, the HT of $c_{\mathrm{i}}(t)$ yields

$$x(t) = \mathrm{Re}\sum_{i=1}^{n} a_i(t)e^{j\phi_i(t)} = \mathrm{Re}\sum_{i=1}^{n} a_i(t)e^{j\int \omega_i(t)dt} \tag{2}$$

where $a_i(t)$ is the instantaneous amplitude of $x(t)$, and $\phi_i(t)$ is the instantaneous phase of $x(t)$. The instantaneous frequency is denoted as

$$\omega_i(t) = \frac{d\phi(t)}{dt} \tag{3}$$

In the polar coordinates system, $x(t)$ is expressed by

$$x(t) = \mathrm{Re}\left(\sum_{i=1}^{n} a_i(t)\exp[j\int \omega_i(t)dt]\right) + r_n(t). \tag{4}$$

Practically, the residue $r_{\mathrm{n}}(t)$ can be ignored.

Let $a_i(\omega, t)$ be the combination of the amplitude $a_i(t)$ and the instantaneous frequency $\omega_i(t)$ of the ith IMF. The HHT of $x(t)$ is given by

$$\mathrm{HHT}(\omega, t) = \sum_{i=1}^{n} a_i(\omega, t). \tag{5}$$

3 Hilbert Spectrum via MODWPT

Analogous to§2 we use EMD to produce monocomponent separation, only here we use the discrete wavelet transform (DWT). Assume we have sampled a continuous–time signal at intervals $\square t = 1$ to a sequence of observation $X = [X_0, X_1, \ldots, X_{N-1}]$ and N is a power of 2. For the class of discrete compactly supported Daubechies wavelets (Daubechies, 1992, Chapter 6) we denote the scaling (low–pass) filter by $\{g_l: l = 0, \ldots, L-1\}$ and the wavelet (high–pass) filter $\{h_l: l = 0, \ldots, L-1\}$. These even–length filters satisfies

$$\sum_{l=0}^{L-1} g_l^2 = 1, \quad \sum_{l=0}^{L-1} g_l g_{l+2n} = \sum_{l=-\infty}^{\infty} g_l g_{l+2n} = 0 \tag{6}$$

for all non–zero integers n, and are related by being quadrature mirror filters:

$$h_l = (-1)^l g_{L-l-1} \text{ or } g_l = (-1)^{l+1} h_{L-l-1} \text{ for } l = 0, \ldots, L-1. \tag{7}$$

For $t = 0, \ldots, N-1$, the jth level wavelet and scaling coefficients are given by

$$V_{j,t} = \sum_{l=0}^{l-1} g_l V_{j-1,(2t+1-l) \bmod N_j - 1} \ (t = 0, \ldots, N_j - 1) \tag{8}$$

$$W_{j,t} = \sum_{l=0}^{l-1} h_l V_{j-1,(2t+1-l) \bmod N_j - 1} \ (t = 0, \ldots, N_j - 1) \tag{9}$$

where mod means modulus after division.

The maximal overlap discrete wavelet transform (MODWT) can be considered as a revised version of the discrete wavelet transform [7]. As previously mentioned, the DWT of level j restricts the sample size to a power of 2, however the MODWT of level j is well defined for any sample size. To conserve energy, we define

$$\tilde{g}_l = g_l / \sqrt{2} \text{ and } \tilde{h}_l = h_l / \sqrt{2} \tag{10}$$

thus, (6) can be transferred to

$$\sum_{l=0}^{L-1} \tilde{g}_l^2 = 1/2, \ \sum_{l=0}^{L-1} \tilde{g}_l \tilde{g}_{l+2n} = \sum_{l=-\infty}^{\infty} \tilde{g}_l \tilde{g}_{l+2n} = 0 \tag{11}$$

and the quadrature mirror filters are defined likewise

$$\tilde{h}_l = (-1)^l \tilde{g}_{L-l-1} \text{ or } \tilde{g}_l = (-1)^{l+1} \tilde{h}_{L-l-1} \text{ for } l = 0, \ldots, L-1. \tag{12}$$

The MODWT creates new filters at each stage by inserting $2^{j-1} - 1$ zeros between the elements of { $\tilde{g}_l$ } and { $\tilde{h}_l$ } to avoid downsampling.

Then, the MODWT pyramid algorithm generates the MODWT wavelet coefficients { $W_{j,t}^{(M)}$ } and the scaling coefficients { $V_{j,t}^{(M)}$ } respectively

$$V_{j,t} = \sum_{l=0}^{l-1} \tilde{g}_l V_{j-1,(t-2^{j-1}l) \bmod N} \ (t = 0, \ldots, N_j - 1) \tag{13}$$

$$W_{j,t} = \sum_{l=0}^{l-1} h_l V_{j-1,(t-2^{j-1}l) \bmod N} \ (t = 0, \ldots, N_j - 1) \tag{14}$$

For the purpose of improving poor resolution at high frequencies, MODWPT is introduced to signal processing. The coefficients at level j and frequency–index n can be expressed as $W_{j,n} = \{W_{j,n,t}, t = 0, \ldots, N-1\}$, and then we produce $\{W_{j,n,t}\}$ using

$$W_{j,n,t} = \sum_{l=0}^{l-1} \tilde{f}_{n,l} W_{j-1,[n/2],(t-2^{j-1}l) \bmod N} \ (t = 0, \ldots, N_j - 1) \tag{15}$$

when n mod 4 = 0 or 3, $\tilde{f}_{n,l} = \tilde{g}_l$; when n mod 4 = 1 or 2, $\tilde{f}_{n,l} = \tilde{h}_l$.

For any signal, the analytic form can be represented as

$$s(t) = W_{j,n}(t) + jH[W_{j,n}(t)]. \tag{16}$$

Then the instantaneous amplitude is denoted by

$$a_{j,n}(t) = \sqrt{W_{j,n}^2(t) + H^2[W_{j,n}(t)]}, \tag{17}$$

and the instantaneous phase function is

$$\phi_{j,n}(t) = tg^{-1} \frac{H[W_{j,n}(t)]}{W_{j,n}(t)} \tag{18}$$

Accordingly, the instantaneous frequency is

$$f_{j,n}(t) = \frac{1}{2\pi}\phi'_{j,n}(t) \tag{19}$$

4 Multitaper Time–Frequency Reassignment

For a nonstationary process $\{x(t), t \in \mathbb{R}\}$, the first definition of its spectrum stands the Wigner–Ville Spectrum (WVS), whose definition is:

$$\mathbf{W}_x(t,f) = \int_{-\infty}^{+\infty} \mathbb{E}\{x(t+\frac{\tau}{2})x^*(t-\frac{\tau}{2})\}e^{-i2\pi f\tau}d\tau. \tag{20}$$

where t and f refer to time and frequency, and $\mathbb{E}\{.\}$ represents the expectation operator. WVD is defined as:

$$W_x(t,f) = \int_{-\infty}^{+\infty} x(t+\frac{\tau}{2})x^*(t-\frac{\tau}{2})e^{-i2\pi f\tau}d\tau. \tag{21}$$

It shows in [13] that, the WVS of a process can be considered as the ensemble average of the WVDs under mild conditions.

Assuming $x(t)$ a form of local stationarity in both time and frequency, the WVS can be estimated as a substitute for the ensemble average of the WVDs. The assumption can be realized by introducing a TF smoothing kernel $\Pi(t,f)$. Thus, the WVS can be estimated as:

$$\hat{\mathbf{W}}_{\mathbf{x}}(t,f) = \int\int_{-\infty}^{+\infty} W_x(s,\xi)\Pi(s-t,\xi-f)dsd\xi = C_x(t,f;\Pi). \tag{22}$$

Compared with the WVD (21), the smoothing kernel $\Pi(t,f)$ in WVS (20) will reduce the cross terms but also lead to a new trade–off between fluctuations and localization. To find a way out, two refinements offered by reassignment and multitapering are combined to ameliorate the contradictory issues of fluctuations reduction and localization.

4.1 Reassignment

Reassignment is a nonlinear technique that is an efficient means of getting sharply localized TF distribution [8][14]. The spectrogram [11] of a signal $x(t)$ with window $h(t)$, which is usually defined as:

$$S_x^{(h)}(t,f) = |F_x^{(h)}(t,f)|^2 = |\int_{-\infty}^{+\infty} x(s)h(s-t)e^{-i2\pi fs}ds|^2, \tag{23}$$

where $F_x^{(h)}(t,f)$ stands for STFT.

Another interpretation of the spectrogram could be a smoothed WVD rather than as a squared STFT:

$$S_x^{(h)}(t,f)=C_x(t,f;W_h), \tag{24}$$

$$C_x(t,f;W_h)=\iint_{-\infty}^{+\infty}W_x(s,\xi)W_h(s-t,\xi-f)dsd\xi \tag{25}$$

with the smoothing kernel $\Pi(t,f)$ in (22), the WVD is supposed to be well localized in both time and frequency. Thus, it gives the clue for improving upon the localization limitations.

Reasoning by a mechanical analogy identifying energy with mass, the core concept of the ‘reassignment’ is to find the center of mass within a domain, then replace it with one single number assigned to the geometrical center of the domain.

On the basis of the previous studies of Auger and Flandrin [8][14], the evaluation of the local centers of mass is defined as,

$$\begin{cases}\hat{t}_{t,f}=t+\mathrm{Re}\{F_x^{(\mathsf{T}h)}(t,f)\,/\,F_x^{(h)}(t,f)\},\\ \hat{f}_{t,f}=f-\mathrm{Im}\{F_x^{(\mathsf{D}h)}(t,f)\,/\,F_x^{(h)}(t,f)\},\end{cases} \tag{26}$$

where the two additional windows needed in the computation are defined from the mother window $h(t)$ as $(\mathsf{T}h)(t) = t\ h(t)$ and $(\mathsf{D}h)(t) = (dh/dt)(t)$. Given the field of all centroids above, the reassigned spectrogram $RS_x^{(h)}(t,f)$ attached to the conventional spectrogram $S_x^{(h)}(t,f)$ follows as:

$$RS_x^{(h)}(t,f)=\iint_{-\infty}^{+\infty}RS_x^{(h)}(s,\xi)\delta(t-\hat{t}_{s,\xi})\delta(f-\hat{f}_{x,\xi})dsd\xi. \tag{27}$$

4.2 Multitapers

Thomson suggested [9] a powerful multiple window spectrum estimator to improve the Welch one, called multitaper, which is to still average the squared Fourier transform (SFTs) stemming from uncorrelated sequences in order to reduce variance, but to construct such sequences by using for each of them the whole data set so as to not sacrifice bias. It can be written as:

$$\hat{S}_{x,K}^{(T)}(f)=\frac{1}{K}\sum_{K=1}^{K}S_x^{(h_k)}(0,f), \tag{28}$$

with a family of basis functions $\{h_k(t), k=1,\ldots,K\}$ and the number of tapers K extending over $(-T/2, +T/2)$.

4.3 Combination and Implementation

According to [15], the expression (28) can be thought of as a WVS estimator:

$$RS_{x,K}(t,f)=\frac{1}{K}\sum_{K=1}^{K}RS_x^{(h_k)}(t,f), \tag{29}$$

with the HF showed in [16], which is defined as

$$h_k(t) = (-1)^k \Big/ \sqrt{\pi^{1/2} 2^k k!}\, g(t)(\mathsf{D}^k \gamma)(t), \tag{30}$$

where $g(t) = e^{-t^2/2}$ and $\gamma(t) = g(it\sqrt{2}) = e^{t^2}$.

In practical, HFs can be computed recursively according to $h_k(t) = H_k(t)g(t) / \sqrt{\pi^{1/2} 2^k k!}$. The $\{H_k(t), k \in N\}$ stands for the Hermite polynomials as below:

$$H_k(t) = 2tH_{k-1}(t) - 2(k-2)H_{k-2}(t), k \geq 2, \tag{31}$$

with the initialization $H_0(t) = 1$ and $H_1(t) = 2t$.

It leads a solution that obeys the recursion below to evaluating the two additional windows $(\mathsf{T}h)(t)$ and $(\mathsf{D}h)(t)$ numerically [11]:

$$(\mathsf{D}h_k)(t) = (\mathsf{T}h_k)(t) - \sqrt{2(k+1)}h_{k+1}(t) \tag{32}$$

5 Results and Discussions

In this section we will compare the performance of the HHT, MODWPT and multitaper TF reassignment method respectively.

5.1 Linear Chirps

In order to test the efficiency and reliability in TF localization, one simple case to consider first is concerned with a monocomponent noise–free chirp signal (512 points, sampling frequency f_s = 1Hz) and another case of single chirp signal embedded in a bandpass noise (with bandwidth = B = 0.2 ~ 0.3Hz, central frequency f_c = 0.25Hz).

Fig. 1 illustrates three methods of TF estimating mentioned above. For noise–free single chirp signal, the frequency of HHT spectrum in diagram (a), although provide the localization of the chirp, disperses from 10s to 110s, and emerges some meaningless frequency from 0.1Hz to 0.2Hz round 90s. The spectrogram of MODWPT in diagram (b) depicted the noise–free chip signal relatively clear. It is noticeable that oscillations and ruptures appear between every two–frequency band of the MODWPT, since the MODWT imposes a fixed octave band tiling on the time–frequency plane. Thus the frequency appears discontinuousness between the segments of frequency intervals. In diagram (c), the estimate of noise–free chip signal based on multitaper TF reassignment offers perfect localization and stability both in time and frequency.

Considering the situation of a monocomponent chirp signal embedded in a bandpass noise with SNR = 10 dB, the ideal estimate should be constant over this band, zero outside and perfectly localized along the chirp instantaneous frequency. Affected by oscillations produced in the process of mode decomposing, the resolution of HHT in diagram (d) hence is not satisfactory, and moreover, the frequency band of the noise is not temporally successive. Temporarily leaving the frequency

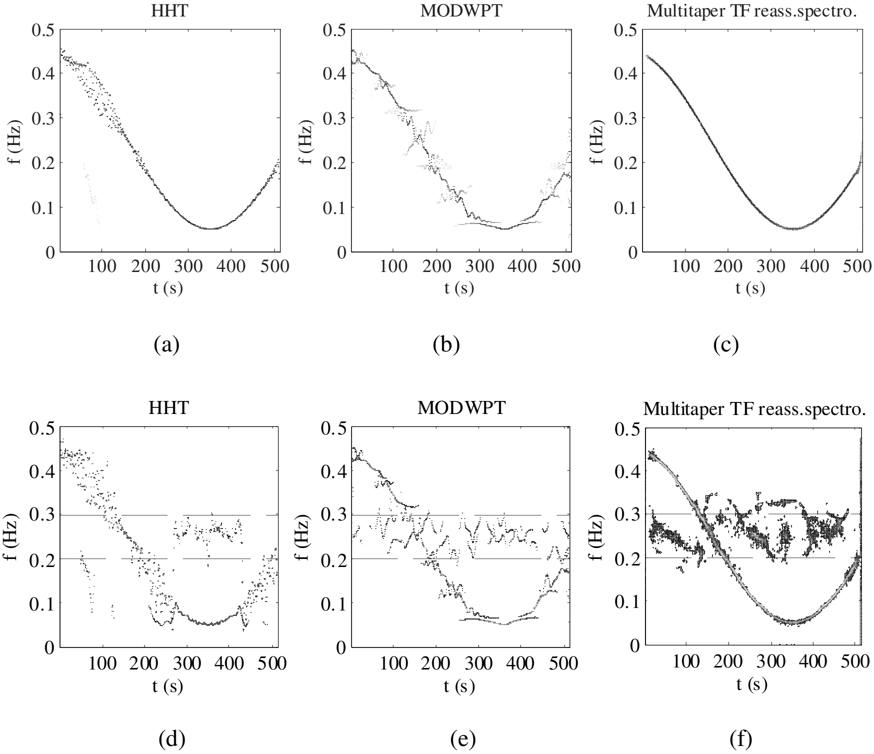
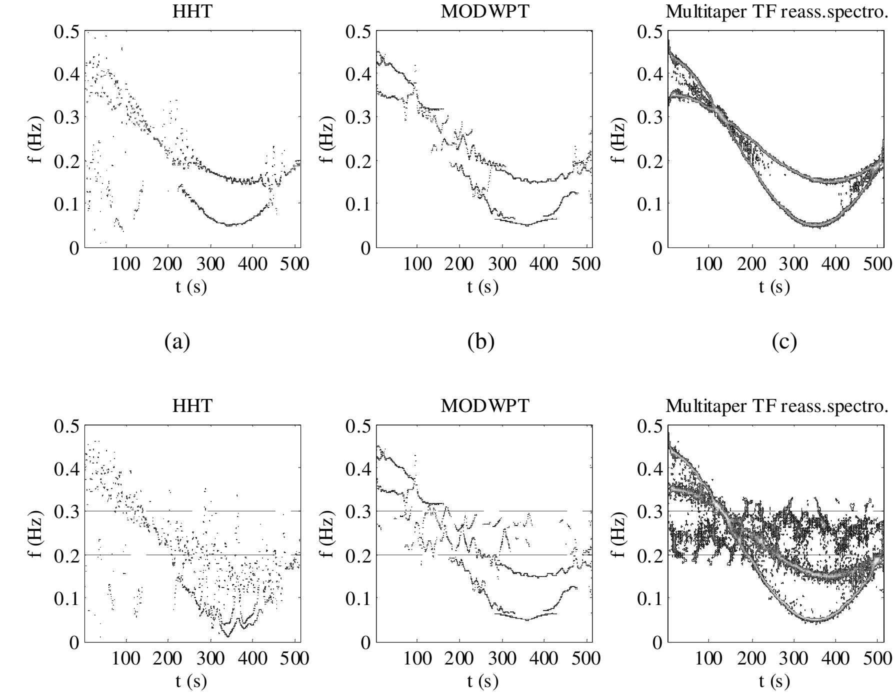


Fig. 1. Comparison of chirp signal TF estimates in case of noise–free and noise (SNR = 10 dB) embedded: (a) HHT of noise–free chirp signal, (b) Hilbert spectrum via MODWPT of noise–free chirp signal at level 3, (c) Multitaper TF reassignment spectrum of noise–free chirp signal with taper length = 95 points (6 Hermite functions), (d) HHT of chirp signal embedded in a bandpass noise (bandwidth = 0.2 ~ 0.3Hz, central frequency f_c = 0.25Hz) denoted by dashed line, (e) the corresponding Hilbert spectrum based on MODWPT of a bandpass noise embedded chirp signal at level 3, (f) Multitaper TF reassignment spectrum of a bandpass noise embedded chirp signal (taper length = 95 points, 6 Hermite functions)

discontinuousness out of account, the MODWPT in diagram (e) outperforms the HHT by providing a perfect localization of the frequency band of the noise. But it suffers statistical fluctuations in the noise band. Relatively, in diagram (f), the multitaper reassigned spectrogram represents the efficiency of combining two techniques, and provides good trade–off between time–frequency localization and smoothness within the bandpass noise.

In Fig. 2, the TF resolution of HHT in diagram (a) is just passable, the intersection of two frequency component of the chirp signal yields illegibility. The performance is awful when chirp signal is embedded the bandpass time–varying noise in the diagram (d). The boundaries between chirp signal and the noise are nearly far from identification. Results of the MODWPT in (b) and (c) are fairly acceptable, although there are still some fluctuations and rupture in frequency domain. The multitaper TF



(a) (b) (c)

(d) (e) (f)

Fig. 2. Comparison of multicomponent chirp signal TF estimates in case of noise–free and noise embedded: (a) HHT of noise–free multicomponent chirp signal, (b) Hilbert spectrum via MODWPT of noise–free multicomponent chirp signal at level 3, (c) Multitaper TF reassignment spectrum of noise–free multicomponent chirp signal with taper length = 95 points (6 Hermite functions), (d) HHT of multicomponent chirp signal embedded in a bandpass noise (bandwidth = 0.2 ~ 0.3Hz, central frequency f_c = 0.25Hz) denoted by dashed line, (e) the corresponding Hilbert spectrum based on MODWPT of a bandpass noise embedded multicomponent chirp signal at level 3, (f) Multitaper TF reassignment spectrum of a bandpass noise embedded multicomponent chirp signal (taper length = 95 points, 6 Hermite functions)

reassigned spectrum acquires favorable TF resolution under the condition of treating two component chirp signal, and also satisfactory when embedded in bandpass noise (SNR = 10 dB).

Fig. 3 illustrates outcomes of three methods in two cases. Ideally, for the bandpass time–limited noise, the estimate should be smooth over the domain of rectangle defined and zero outside. Apparently, in (a) and (d), energy leak appears in the estimate of HHT at low frequency as a result of end effects by EMD. By comparison, MODWPT in (b), otherwise provides well localization in the rectangle domain with slight spreads outside, although still suffers large fluctuations. In terms of decreasing fluctuations and preserving localization, the multitaper TF reassigned spectrogram gives some very good performance and excellent convergency.

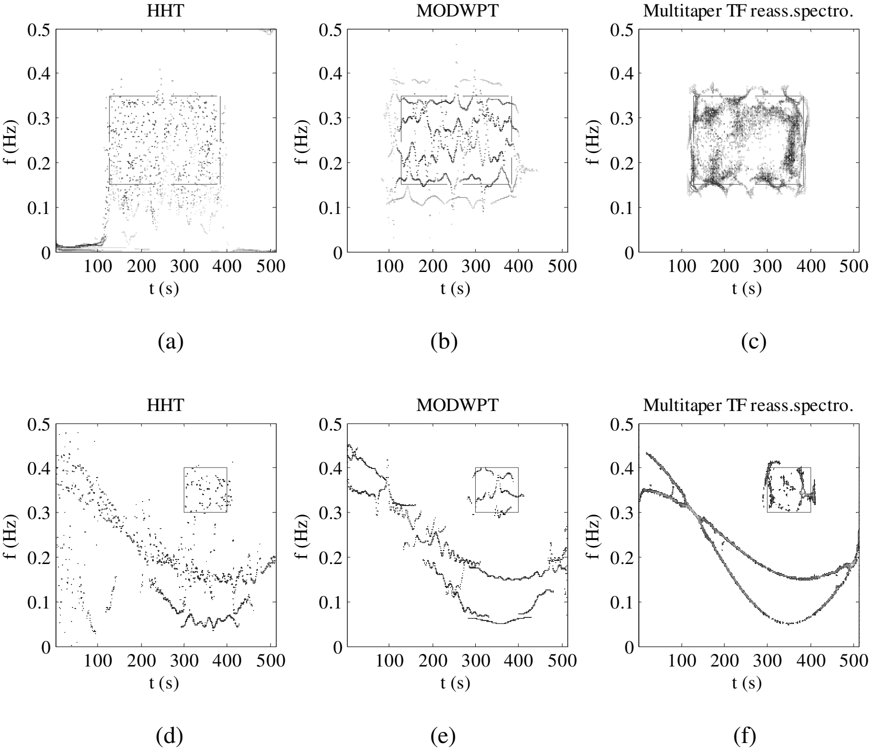


Fig. 3. Comparison of noise estimates: (a) HHT of bandpass time–limited noise (bandwidth = 0.15 ~ 0.35Hz, central frequency f_c = 0.25Hz), (b) Hilbert spectrum via MODWPT of noise at level 3, (c) the corresponding estimate of Multitaper TF reassignment spectrum with taper length = 95 points (10 Hermite functions), (d) HHT of multicomponent chirp signal with a transient bandpass time–limited noise denoted by pane in dashed line, (e) the corresponding Hilbert spectrum based on MODWPT at level 3, (f) Multitaper TF reassignment spectrum of the chirp + noise (taper length = 95 points, 10 Hermite functions)

5.2 Effectiveness Measure by Rényi Entropy

Intuitively speaking, we consider that a component is a concentration of energy in some domain, but it remains difficulties to translate this idea into a quantitative concept [17][18][19]. Time frequency representations (TFRs) generalize the notion of the time and frequency domains to a joint time frequency function $C_x(t, f)$ that demonstrates how the frequency content of a signal x changes over time [20][21]. Document [22] suggests the classical Shannon entropy (given below)

$$H(C_x) := -\iint C_x(t,f)\log_2 C_x(t,f)dtdf \ , \tag{33}$$

as a natural candidate for estimating the complexity of a signal through its TFR. However, the application of the Shannon entropy is unfortunately hindered due to the negative values taken on by most TFRs, as in Eq. (33).

Williams, Brown, and Hero sidestepped the negativity problem by using the generalized entropies of Rényi [23].

$$H_\alpha(C_x) = \frac{1}{1-\alpha}\log_2 \iint C_x^\alpha(t,f)dtdf \tag{34}$$

where $\alpha > 0$.

For any discrete signal, the joint time frequency function $C_x(t, f)$ can be normalized as a discrete TF distribution $P(n,m)$, then a Rényi entropies generates as below [23]

$$H_\alpha(P) = \frac{1}{1-\alpha}\log_2 \sum_{n=1}^{N}\sum_{m=1}^{M}(P(n,m))^\alpha, \alpha > 0 \tag{35}$$

where $P(n,m) = C_x[n,m] / \sum_{n'=1}^{N'}\sum_{m'=1}^{M'} C_x[n',m']$.

The effectiveness of three TF methods referred in former sections is charted in Table. 1 that measured by the Rényi Entropy in the case of linear chirp and of a Gaussian noise (512 points each). As it indicates in [11], we have ideally $P_{chirp}(n,m) = \delta_{n,m} / N$ and $H_\alpha(P_{chirp}) = \log_2 N$ for the linear chirp, and by contrast, it leads $P_{noise}(n,m) = 1/N^2$ and $H_\alpha(P_{noise}) = 2\log_2 N$ for a white Gaussian noise. Considering the case $\alpha = 5$, the ideal Rényi Entropy of chirp is H_5 = 9 and that of noise becomes H_5 = 18. The results are given below

Table 1. Rényi Entropy

METHOD	Linear Chirp	white Gaussian Noise
HHT	9.3122	10.2624
MODWPT	9.5381	11.2866
Multitaper TF reass. (10 tapers)	9.7999	13.4014

A practical example of bat–echolocation signal is discussed in Fig.5. It contains a cluster of high frequency pulses uttered by bats. The estimate by HHT is unreadable that aliasing still remains. The MODWPT identifies the four high frequency components successfully, although with fluctuations between layers of wavelet. It is clearly shown in (c) that the multitaper TF reassigned approach not only pulls out all of the four actual frequency components, but also provides perfect stability in time and in frequency domain.

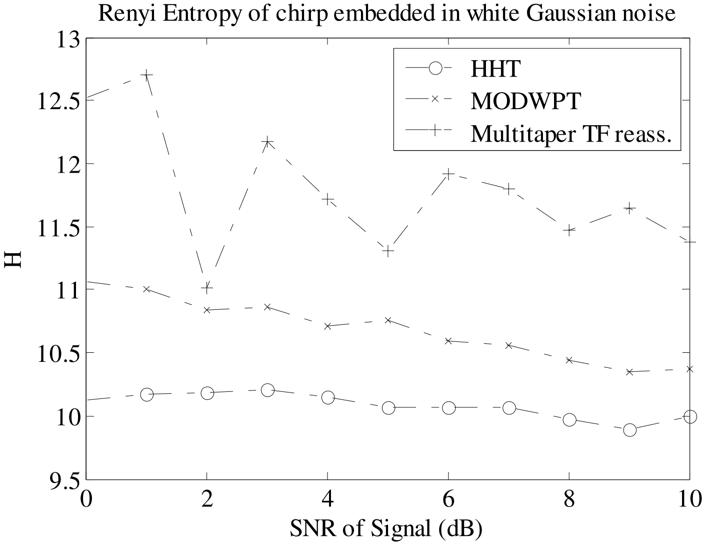
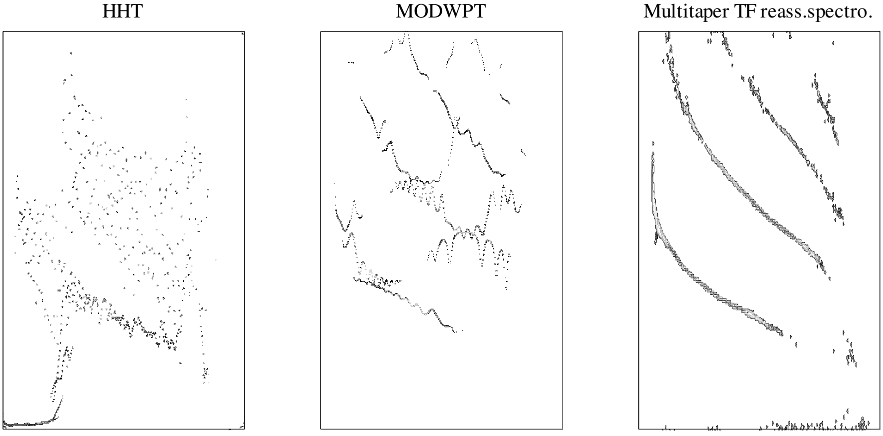


Fig. 4. Comparison of Rényi Entropy: Rényi Entropy comparison of three TFR methods of a linear chirp in various levels of noise (512 points each)



(a) (b) (c)

Fig. 5. Comparison of TF estimates: bat–echolocation signal: Each diagram respectively represents the estimates based on HHT, MODWPT, and multitaper TF reassigned spectrogram (10 Hermite functions)

5 Conclusions

Three noteworthy approaches of TF estimating have been compared, with several cases such as a single chirp signal noise–free and noise–embedded, multicomponent chirp signal noise–free and noise–embedded. The resolution of each representation in time and frequency domain is illustrated and the inherent drawbacks of them are indicated. By avoiding several shortcomings of HHT, the MODWPT will probably become an appropriate method for nonstationary signal analysis. On the purpose for decreasing fluctuations while preserving localization, the multitaper TF reassigned spectrogram gives outstanding performance for the signals enumerated. Due to space limitation, ameliorations of each method, not been discussed in this paper, are under investigation and will be reported elsewhere.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under the project grant numbers 60573125 and 60873264.

References

1. Gabor, D.: Theory of communication. J. Inst. Electron. Eng. 93(11), 429–457 (1946)
2. Allen, J.B., Rl Rabiner, L.: A unified approach to short–time Fourier analysis and synthesis. Proc. IEEE. 65, 1558–1566 (1977)
3. Huang, N.E., Shen, Z., Long, S.R.: A new view of nonlinear water waves: the Hilbert spectrum. Annual Review of Fluid Mechanics 31, 417–457 (1999)
4. Datig, M., Schlurmann, T.: Performance and limitations of the Hilbert–Huang transformation (HHT) with an application to irregular water waves. Ocean Engineering 31(14), 1783–1834 (2004)
5. Jingping, Z., Daji, H.: Mirror extending and circular spline function for empirical mode decomposition method. Journal of Zhejiang University (Science) 2(3), 247–252 (2001)
6. Walden, A.T., Contreras, C.A.: The phase–corrected undecimated discrete wavelet packet transform and its application to interpreting the timing of events. Proceedings of the Royal Society of London Series 454, 2243–2266 (1998)
7. Tsakiroglou, E., Walden, A.T.: From Blackman–Tukey pilot estimators to wavelet packet estimators: a modern perspective on an old spectrum estimation idea. Signal Processing 82, 1425–1441 (2002)
8. Auger, F., Flandrin, P.: Improving the readability of Time–Frequency and Time–Scale representations by the reassignment method. IEEE Transactions on Signal Processing 43(5), 1068–1089 (1995)
9. Thomson, D.J.: Spectrum estimation and harmonic analysis. Proc. IEEE 70, 1055–1096 (1982)
10. Bayram, M., Baraniuk, R.G.: Multiple Window Time–Frequency Analysis. In: Proc. IEEE Int. Symp. Time–Frequency and Time–Scale Analysis (May 1996)
11. Xiao, J., Flandrin, P.: Multitaper time–frequency reassignment for nonstationary spectrum estimation and chirp enhancement. IEEE Trans. Sig. Proc. 55(6) (Part 2), 2851–2860 (2007)

12. Huang, N.E., Shen, Z., Long, S.R., Wu, M.L., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H.: The Empirical Mode Decomposition and Hilbert Spectrum for Nonlinear and Non–Stationary Time Series Analysis. Proc. Roy. Soc. London A 454, 903–995 (1998)
13. Flandrin, P.: Time Frequency/Time Scale Analysis. Academic Press, London (1999)
14. Flandrin, P., Auger, F., Chassande–Mottin, E.: Time Frequency Reassignment From Principles to Algorithms. In: Papandreou–Suppappola, A. (ed.) Applications in Time Frequency Signal Processing, vol. 5, pp. 179–203. CRC Press, Boca Raton (2003)
15. Boashash, B.: Time frequency signal analysis and processing: a comprehensive reference. Elsevier, London (2003)
16. Bayram, M., Baraniuk, R.G.: Multiple window time varying spectrum estimation. In: Fitzgerald, W.J., et al. (eds.) Nonlinear and Nonstationary Signal Processing, pp. 292–316. Cambridge Univ. Press, Cambridge (2000)
17. Williams, W.J., Brown, M.L., Hero, A.O.: Uncertainty, information, and time frequency distributions. In: Proc. SPIE Int. Soc. Opt. Eng., vol. 1566, pp. 144–156 (1991)
18. Orr, R.: Dimensionality of signal sets. In: Proc. SPIE Int. Soc. Opt. Eng., vol. 1565, pp. 435–446 (1991)
19. Cohen, L.: What is a multicomponent signal? In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing ICASSP 1992, vol. V, pp. 113–116 (1992)
20. Cohen, L.: Time Frequency Analysis. Prentice–Hall, Englewood Cliffs (1995)
21. Baraniuk, R.G., Flandrin, P., Janssen, A.J.E.M., Michel, O.: Measuring time frequency information content using the Renyi Entropies. IEEE Transactions on Information Theory 47(4), 1391–1409 (2007)
22. Shannon, C.E.: A mathematical theory of communication, Part I. Bell Sys. Tech. J. 27, 379–423 (1948)
23. Rényi, A.: On measures of entropy and information. In: Proc. 4th Berkeley Symp. Math. Stat. And Prob., vol. 1, pp. 547–561 (1961)

Performance Analysis of Greenhouses with Integrated Photovoltaic Modules

Maurizio Carlini[1], Mauro Villarini[2], Stefano Esposto[2], and Milena Bernardi[2]

[1] Facoltà di Agraria, Università della Tuscia, Via S. Camillo de Lellis, 01100 Viterbo Italy
[2] CIRPS Sapienza Università di Roma, piazza San Pietro in Vincoli, 10 - 00184 Rome Italy

Abstract. Thanks to the DM 19.02.2007, Italian government supported the development and the expansion of solar photovoltaic in Italy. The feed-in tariff had a great success, and like in Spain and Germany big size photovoltaic plants have been built, especially in the south of the country. The south of Italy presents high irradiation (up to 1.700 equivalent hours) and economically agriculture is an important local resource. This aspect led to the concept of the solar greenhouses, a way to match the electricity production by PV modules with an improvement of the cultivation possibilities. Solar greenhouses includes integrated PV modules mounted on the roof oriented to south and his design is new and still has to be evaluated in details. In particular important parameters like the luminance, the type of cultivations and the temperature of the PV modules must be carefully analyzed to have a real good match between the agriculture and the electricity production purpose. In the paper TRNSYS 16 has been used for the simulation of temperatures and humidity in the structure. The simulation had the goal to define the performance of the solar greenhouse during the year, with the possibility to individuate important construction parameters for the realization of a greenhouse efficient from all point of views.

Keywords: Solar greenhouse, PV modules, TRNSYS 16 Simulation.

1 Introduction

The electricity produced by photovoltaic modules on greenhouse's coverage can be used for the same greenhouse: for the air-conditioning systems, for its control and management, irrigation, ventilation, material handling and so on. The idea to realize some greenhouses that, thanks to PV modules used as coverage, are able to produce electricity, solves the problem of the high energetic consumption for the agriculture production, but at the same time it makes even more complex the argument in relation to the design. In fact even though they can produce electricity through integration with PV modules, the greenhouse must maintain their principal purpose: allow the agricultural cultivations. Therefore the crops, and accordingly the conditions they require, are instrumental in define the design parameters. The integration of PV modules on the roof determines changes in normal condition inside the greenhouse; the changes involves both thermal property, temperature and humidity, and the condition of luminance. In fact the crops that are cultivated in protected environments require particular climatic conditions and it can happen that the PV modules

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 206–214, 2010.

determine not optimal condition. The solar greenhouses must therefore be designed to ensure the right balance between an adequate internal radiation in the winter and adequate protection from excessive radiation during the summer. The brightness of the greenhouse is fundamental for crops; must be uniform as possible and more guaranteed to the greatest number of daily times. Because of the complexity of the system it is necessary study the configuration through a multidisciplinary approach.

1.1 Background and Method

The integration of PV modules in greenhouses structures is not common, and reliable field data are not present in literature yet. The topic became important in the PV sector because the Italian PV incentive feed-in tariff acknowledge a higher tariff for the integrated solutions, including the greenhouses. The agricultural sector had the opportunity to enter deeply in the growing market of the photovoltaic in Italy, and many investors are elaborating mixed business plan cultivation/electricity production. However, many technical considerations have to be taken in consideration to have a successful solar greenhouse. In particular must be considered the fact that the southern part of the greenhouse coverage is "blackened" by the presence of the modules, reducing considerably the light available for the cultivations. New types of modules which foresee transparent parts between the cells to improve transparency of the whole module are under patent, the problem is linked to the official certification of these technical solutions which is still missing. The main indicator used in cultivations to evaluate the light need of a specific species is the illuminance, expressed in lux. The software Relux 2007 is normally used for the evaluation of the light availability in resident buildings and offices, and it calculates the spatial value of illuminance.

The analysis is regarding the thermal performance of the solar greenhouse. The completely closed greenhouse has the goal to maintain high temperature during the winter to allow specific cultivations all over the year. This aspect of the greenhouse use does not match with the requirement of cold surfaces for having high efficiency in the conversion light/electricity in the PV modules. The thermal losses are the main ones to affect the production of PV modules and some systems are proposed to cool off the back surface of the integrated modules, using air ventilation o water to bring down the temperature. This aspect complicates the system and the paper evaluates the impact of such systems on the overall production. To simulate the different scenario, the software TRNSYS 16 has been used. The greenhouse has been modeled in the software and the performance all over the year gave results in terms of average temperature and humidity inside the greenhouse, for agriculture purpose, and the back temperature of the PV modules, for electricity production purpose.

1.2 Italian Background and Law References

In Italy GSE is the company that recognizes and supplies the benefit for the electricity production from renewable Energy installation, among which solar photovoltaic. The tariff recognized by mechanism of "Conto Energia" are diversified on the basis of the rated capacity and architectural integration. In particular the solar greenhouse are a part of the category "Totally integrated" and because of this GSE acknowledge to solar greenhouse the higher tariff for the determined class of capacity. To be able to reenter in category "totally integrated" it is necessary that: " PV modules are constructive elements of the coverage or the walls of the manufact used as greenhouse in which agricultural cultivations or floriculture are cultivated, permanently, for all the years of acknowledgement of the benefit. The structure of the greenhouse, in metal, wood or masonry, must be closed (the closing can seasonally be eventually removable), fixed and anchored to the ground. The PV modules, the portion of coverage of the greenhouse in which they are integrated, must have a distance lowest distance from the ground of 2 metres." This definition is presented in "book of architectural integration" by GSE and later on in this article we present this type of installation. Protected crops in Italy are, from economic point of view, very important, both for their high extension (more than 27.000 ha, according to ISTAT 2000 sources) for internal production and export of vegetable and fruit products (about 23.000 ha) and flowers products (4.000 ha).

2 TRNSYS 16 Simulation

TRNSYS 16 is used for the evaluation of temperature and humidity in a solar greenhouse located in the south of Italy, in Cassano Jonio, during a typical year; the plant of the greenhouse is rectangular and the orientation and dimensions of walls are:

1) North face: surface 304.35 m2
2) East face: surface 35.73 m2
3) South face: surface 139.82 m2
4) West face: surface 35.73 m2.

The coverage is realized by a metallic structure, tilt 25 °; the PV modules on the coverage are interrupted by 9 Solartubes 21"ceiling. The Solartubes are used to increase internal light.
The material of the walls are:

- Plastic sheet on the north and south face;
- Polycarbonate, thickness 6 mm, on the east and west face.

2.1 Meteorological Data and Material

The meteorological data for the place where the greenhouse is located, (39°47'9.46"N 16°27'20.99"E) are simulated by the software Meteonorm 6.0 . A specific file is created with data relative to 8760 hours, with time step of 1 hour.

The material, plastic and polycarbonate, are not present in standard library of TRNSYS 16. Therefore the file for new material are created with software Optics e Window 5 and the new file are added to the standard library of TRNSYS 16.

2.2 Hypothesis

Hypothesis for the greenhouse model are:

a) Coverage of greenhouse tilted south, to maximize energy production;
b) Polycarbonate has been modelled with Lexan XL, thickness 6mm, transparent;

c) Plastic sheet has been modelled with Plexiglass MC , thickness 2.5 mm, clear plastic;
d) The Solartubes has been modelled as artificial light in greenhouse. Solartubes are used to increase the light available for the cultivations under PV modules. With TRNSYS the Solartubes are evaluated insofar of energy they supply. So from the characteristic of lightness of Solartube 21” ceiling, we have a value of 10833 Lumen during a typical year; considering the ratio lumen/watt equal about to 10, consequently the 9 Solartubes supplies energy in greenhouse equal to 9794.7 W;
e) Air replacement of 40% of volume per hour;
f) Air replacement due only to permeability or loss through plastic sheet.

2.3 Simulation

The structure of the greenhouse has been simulated with TRNBUILD; TRNBUILD provides data files necessary for using the TRNSYS TYPE 56 Multi-Zone Building component in TRNSYS Studio. The simulation has been carried out for all over the year, from 1 January to 31 December.

The simulation gave results in terms of average temperature and humidity inside the greenhouse, for agriculture purpose, and the back temperature of the PV modules, for electricity production purpose.

In fact the temperature of PV modules influences the energy produced; for example crystalline modules, every ten degrees of temperature, have lower capacity of 4-5 %. Usually it is possible to restrict these losses through a displacement that allows the natural ventilation on the back face.

3 Results

The collecting of the results from the simulations led to evaluate the correct cultivations which could be “matched” with solar greenhouses systems. In particular

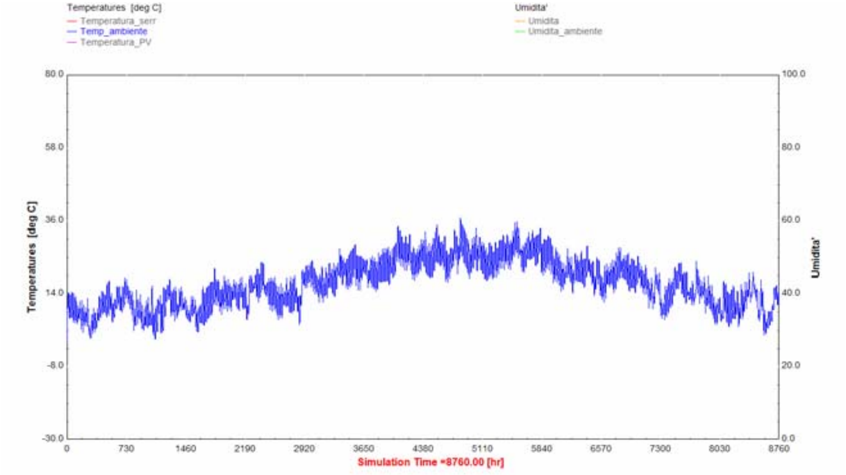


Fig. 1. Ambient temperature (°C), 1 January - 31 December

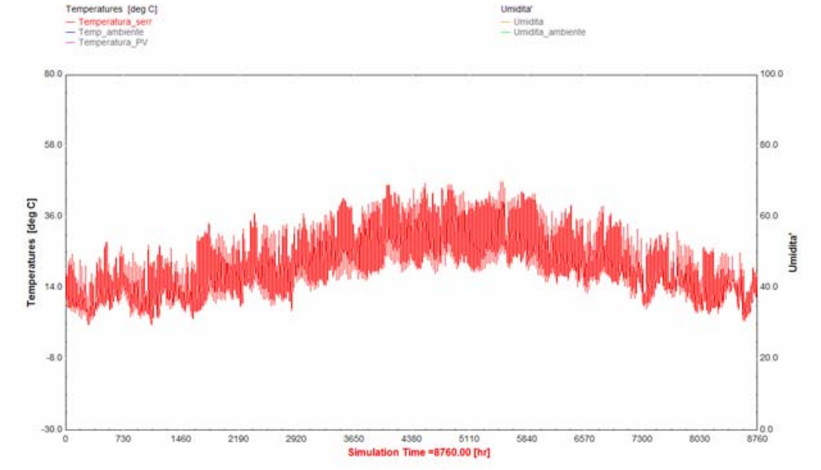


Fig. 2. Solar greenhouse temperature (°C), 1 January - 31 December

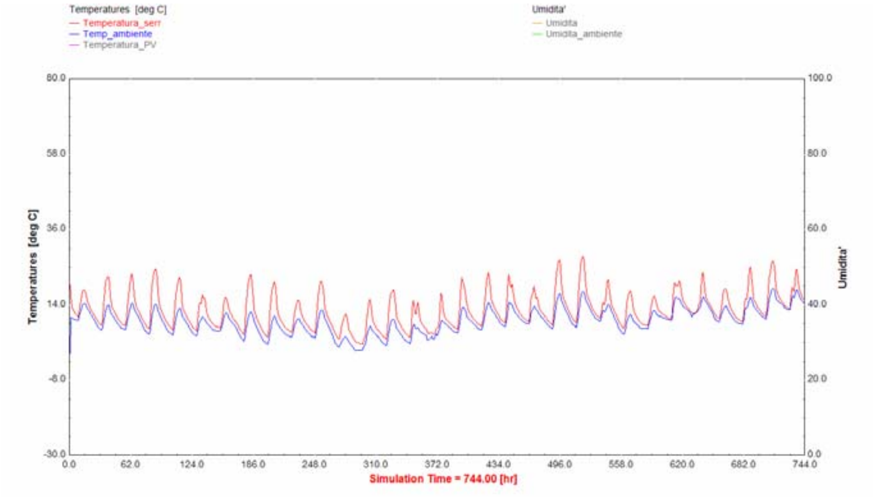


Fig. 3. Ambient temperature (blue line) and Solar greenhouse temperature (°C) (red line), 1 - 31 January

the calculations allowed adapting the normal design of the greenhouse to the new goal, the electricity production. In particular graphs relative to temperature and humidity has been obtained and under represented.

3.1 Temperature

The evolution of temperatures found interesting to analyze are reported in Fig. 1, Fig. 2, Fig. 5. First of all ambient temperature (Fig. 1) has been graphicated, like this

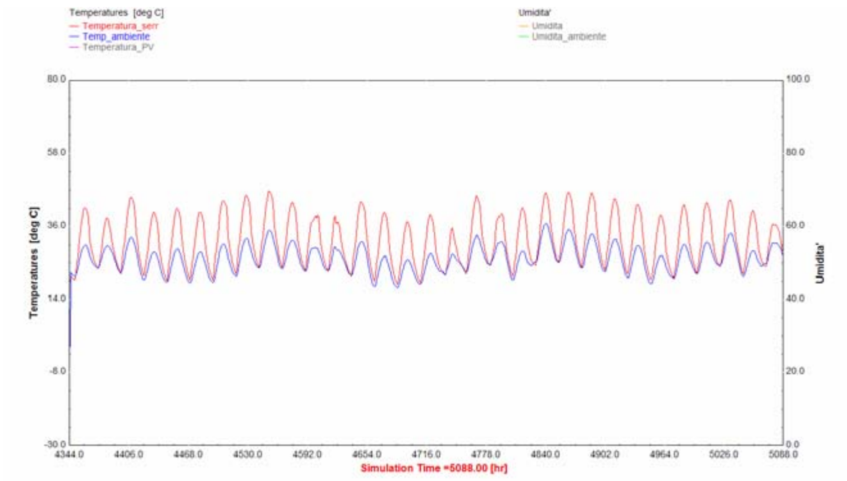


Fig. 4. Ambient temperature (blue line) and Solar greenhouse temperature (°C) (red line), 1 - 31 July

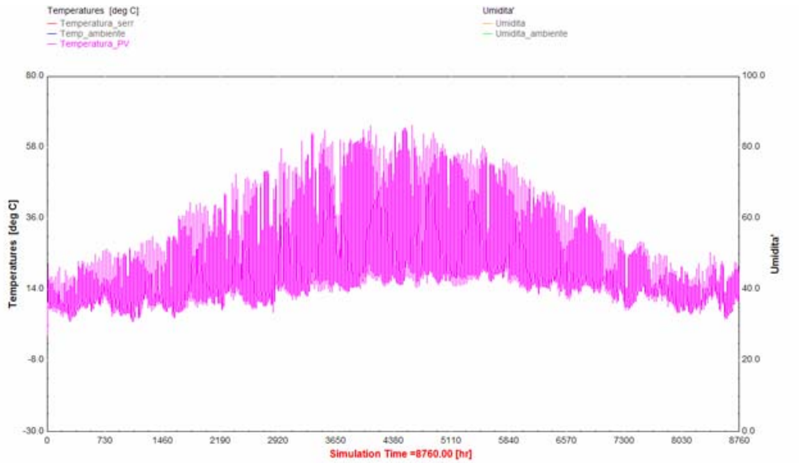


Fig. 5. PV module temperature (°C) – 1 January, 31 December

it is possible make a comparison between the same temperature and the temperature in solar greenhouse (Fig. 2).

The temperature in the greenhouse is greater than ambient temperature about 11°C, but only in the hottest hour of the day; the difference is clearly visible in Fig. 3, where has been graphicated the evolution of temperature only in January, considered representative month of winter season. In the coldest hours, for example at night, the difference between the 2 temperatures is least; so we can say that in cool season during the night, or for example in harsh climate, it is necessary heating system to maintain higher temperatures. However during daily hottest hours, temperatures reached inside of the greenhouse are good for the crops.

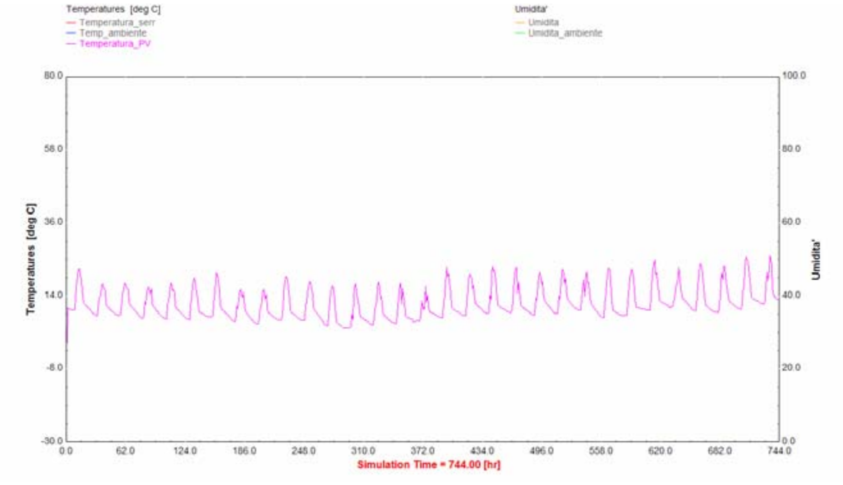


Fig. 6. PV module temperature (°C), 1 – 31 January

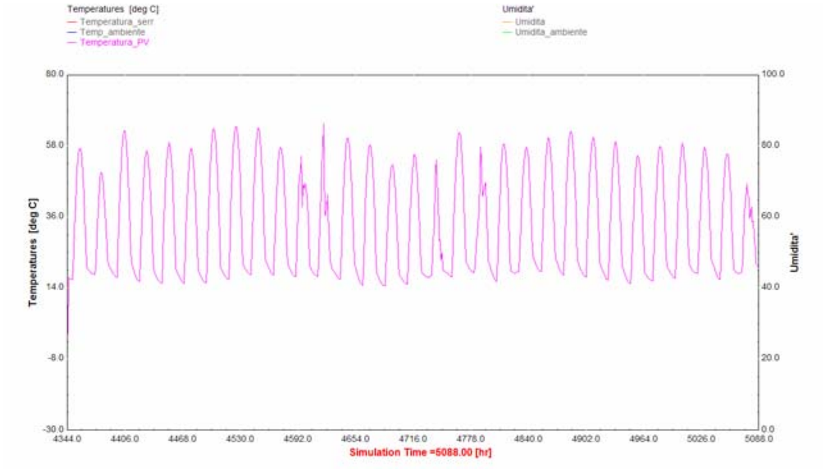


Fig. 7. PV module temperature (°C), 1 – 31 July

The same graph in Fig. 4, ambient temperature- solar greenhouse temperature, is reported for a representative month of summer season (July); during daily hottest hours we can see that the greenhouse temperature reachs too high values; for this reason, in summer, the cultivation in greenhouse is not recommended, and it is possible only with ventilation or shading systems.

In fig. 5 is reported the evolution of photovoltaic modules temperature.

In winter (Fig. 6), considering always January as representative month, the temperature remains relatively low, with no problem for the performance of PV modules; while during summer month (Fig. 7) temperature reaches "dangerous"

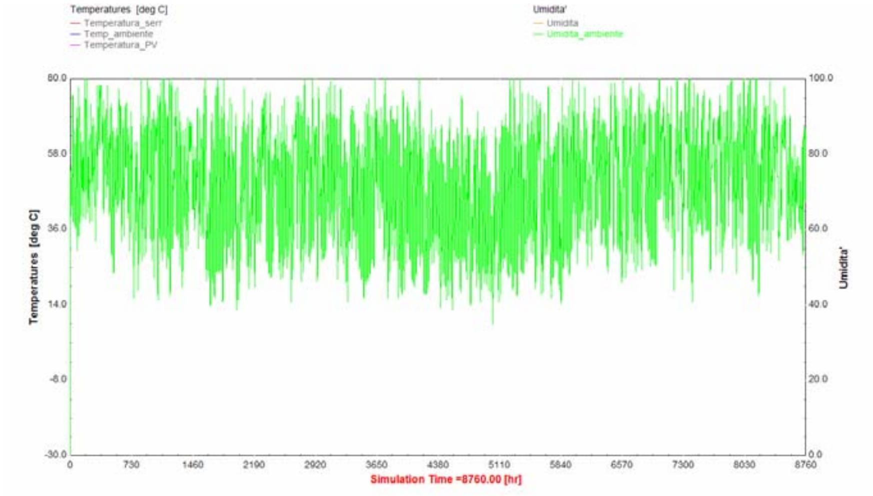


Fig. 8. Ambient humidity, 1 January - 31 December

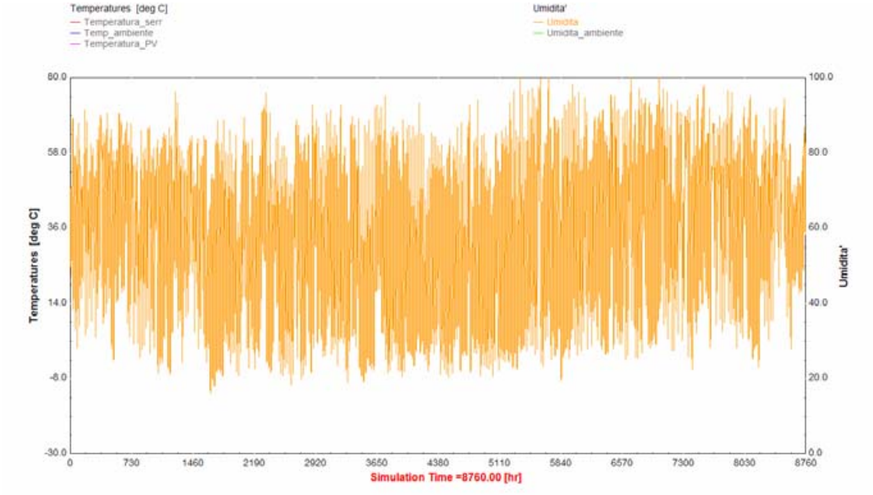


Fig. 9. Solar greenhouse humidity, 1 January - 31 December

values , about up to 60 °C. These temperatures result in decreased performance, approximately 20%.

3.2 Humidity

Relative to humidity in Fig. 8 and in Fig. 9 it can see that the PV coverage produces a slight decrease of Solar greenhouse humidity compare to ambient value, and so an improvement of the conditions for the crops.

4 Conclusions

In this article has been analyzed the thermal behavior of a greenhouse; it is seen its behavior is not the optimal neither in winter not in summer.

Following, it will refer only to the hours of the day light, because of we consider only the combined effect: optimal condition for cultivations and energy production, that verify only during the hours of solar radiation.

During winter season, or when the ambient temperature is low, two condition are true: on the one hand the lowest temperature in greenhouse and the other low PV module loss. However during summer season the temperature in greenhouse is too high, and consequently can be an overheating of PV modules that decrease the performance.

Some systems are proposed to cool off the back surface of the integrated modules, using air ventilation o water to bring down the temperature.

But we can consider that solar radiation is higher in summer than in winter so the contribution of the losses can be decreased from increased solar radiation.

At this point, we can do evaluations only about the internal temperature. As we have seen in summer season the internal temperature of solar greenhouse reaches too high values harmful to crops that are cultivated in Italy.

But, at the same time, during winter night the temperature the temperature undergoes a sudden fall, until to ambient temperature. For this reason it is necessary introduce external systems, heating or cooling or ventilaton, to maintain conditions for crops.

References

1. Nayak, S., Ghosal, M.K., Tiwari, G.N.: Performance of Winter Greenhouse Coupled with Solar Photovoltaic and Earth Air Heat Exchanger. Agricultural Engineering International: the CIGR Ejournal Manuscript EE 07 015 IX (Nvember 2007)
2. Canakci, M., Akinci, I.: Energy use pattern analyses of greenhouse vegetable production. Energy 31, 1243–1256 (2006)
3. Yanoa, A., Furuea, A., Kadowakia, M., Tanakab, T., Hirakib, E., Miyamotob, M., Ishizuc, F., Nodad, S.: Electrical energy generated by photovoltaic modules mounted inside the roof of a north–south oriented greenhouse. Biosystems Engineering 103, 228–238 (2009)
4. Nayak, S., Tiwari, G.N.: Theoretical performance assessment of an integrated photovoltaic and earth air heat exchanger greenhouse using energy and exergy analysis methods. Energy and Buildings 41, 888–896 (2009)
5. Nayak, S., Tiwari, G.N.: Energy and exergy analysis of photovoltaic/thermal integrated with a solar greenhouse. Energy and Buildings 40, 2015–2021 (2008)
6. Janjai, S., Lamlert, N., Intawee, P., Mahayothee, B., Bala, B.K., Nagle, M., Muller, J.: Experimental and simulated performance of a PV-ventilated solar greenhouse dryer for drying of peeled longan and banana. Solar Energy 83, 1550–1565 (2009)
7. Sethia, V.P., Sharma, S.K.: Greenhouse heating and cooling using aquifer water. Energy 32, 1414–1421 (2007)

Modelling Cutaneous Senescence Process

Maria Crisan[1], Carlo Cattani[2], Radu Badea[3], Paulina Mitrea[4], Mira Florea[5],
Diana Crisan[1], Delia Mitrea[4], Razvan Bucur[6], and Gabriela Checiches[6]

[1] Department of Histology, University of Medicine and Pharmacy “Iuliu Hatieganu”,
Cluj- Napoca, Romania
mcrisan7@yahoo.com, diana_crisan2002@yahoo.com
[2] Department of Pharmaceutical Sciences, University of Salerno, Italy
ccattani@unisa.it
[3] Department of Ultrasonography, University of Medicine and Pharmacy “Iuliu Hatieganu”
Cluj- Napoca, Romania
rbadea2003@yahoo.com
[4] Faculty of Automation and Computer Science, Department of Computer Science, Technical
University of Cluj-Napoca, Romania
paulina.mitrea@cs.utcluj.ro, delia.mitrea@cs.utcluj.ro
[5] Mira Florea, Family Medicine Department, Internal medicine speciality , University of
Medicine and Pharmacy “Iuliu Hatieganu” Cluj- Napoca, Romania
[6] Clinic of Dermatology, Regional Emergency Hospital , Cluj-Napoca

Abstract. During the last years, skin aging has become an area of increasing research interest. High frequency ultrasound allows the “in vivo” appreciation of certain histological parameters and offers new characteristic markers, which may quantify the severity of the cutaneous senescence process. This paper focuses on measuring the changes in skin thickness and dermis echogenicity , as part of the complex ageing process, on different intervals of age. In particular by using a multiscale approach we will compute some parameters which are connected with complexity (fractal structure) of skin ageing.

Keywords: Ultrasound; Senescence; Skin modeling.

1 Introduction

Senescence represents a natural, slow and irreversible process, affecting all-tissues of an organism, being determined by numerous factors, each with a different contribution. The aging phenomenon involves intrinsic and extrinsic reactions. It leads to characteristic changes which occur at molecular, cellular, tissular and clinical level [1]. Unlike other organs, the skin is in direct contact with the environment and therefore undergoes chronological and sun-induced aging. The non-invasive assessment of chronological and photoaging cumulative processes still represents a challenge.

Ultrasound is used in dermatology as a non-invasive diagnostic tool in the range of 20-150MHz. High frequency ultrasound allows the “in vivo” appreciation of certain histological parameters and offers new characteristic markers, which may quantify the

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 215–224, 2010.

severity of the cutaneous senescence process [2]. Moreover, it may differentiate between the chronological aging process and photoaging. It evaluates the physio-chemical properties of the integument, epidermis, dermis and subcutis that induce acoustical variations, expressed through certain changes of tissue echogenicity. Several studies have investigated the role of ultrasound as a non - invasive tool for the appreciation of skin ageing, skin lesions, and efficacy of certain therapies [3,4]. Our study focuses on measuring the changes in skin thickness and dermis echogenicity, as part of the complex ageing process, on different intervals of age. It is a part of a larger study in which several noninvasive and biomarker methods are used to appreciate the skin response to ultraviolet rays. We consider that it is one of the most complex studies in this field because of the multiple parameters measured for each subject.

2 Materials and Method

The study was performed on 40 Caucasian patients, 12 men, 28 women, aged between 4 -75 years and divided into four age categories: 4-20, 21-40, 41-60, >60. For each subject, cutaneous ultrasound images were taken from 3 different sites: dorsal forearm (DF), medial arm (MA) and zygomatic area (ZA).

The ultrasound investigation was performed using a high frequency cutaneous ultrasound device DermaScan™ 20 MHz [30] which allowed us to obtain "in vivo" sectional images of the tissue up to 2.5 cm. in depth.

Dermascan consists of three major components: a transducer, an elaboration system and a database [30]. The ultrasonic wave is partially reflected at the interface of adjacent structures, generating areas with different echogenicity amplitudes,

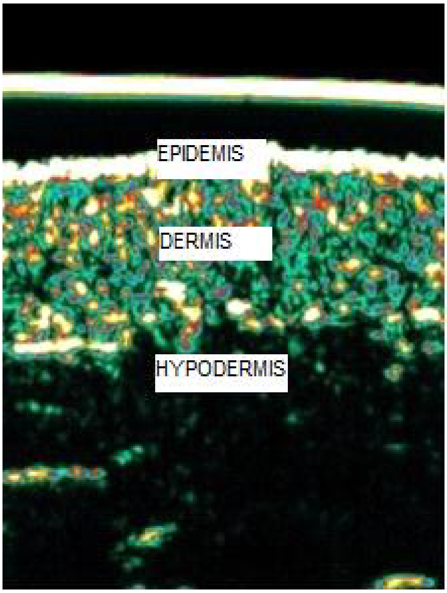


Fig. 1. Ultrasound image of the skin: epidermis, dermis, hypodermis

echogenicity being the ability to create an echo, i.e. return a signal in ultrasound examinations. The intensity of the echogenicity is evaluated by a microprocessor and visualized as a colored bidimensional image. The color scale of the echogenicity is as follows: white>yellow>green>blue>black.

Normally, the epidermal echogenicity appears as a white band, the dermis is expressed as a 2 color composition: yellow and/or red, and the subcutaneous layer appears either green or black (Figure 1).

The gain curve was adjusted at a value of 20 dB, at a speed of ultrasound at tissular level of 1580m/s. [5]

2.1 Image Capture

The images were obtained using the transducer, placed perpendicularly on the site of interest, and then analyzed, using a special software (Dermavision), that allowed the segmentation of each image in different regions of interest, by selecting one or more homogenous amplitude bands.

After having defined an area of interest, by selecting a certain amplitude interval, we assessed quantitatively the areas where the amplitude belonged to the chosen intervals, and obtained its extension both in mm and number of pixels.

2.2 Parameters of Interest

For each image, we measured the following parameters:

1. LEP (0-30), the number of low echogenic pixels of the dermis;
2. MEP (50-100, 100-150), the number of medium echogenic pixels of the dermis;
3. HEP (200-255) the number of high echogenic pixels of the dermis;
4. SLEB, the subepidermal low echogenicity band;
5. LEPs/LEPi the ratio between the number of low echogenic pixels in the upper dermis over the number of low echogenic pixels in the lower dermis.

2.3 Measurement of Low, Medium and High Echogenicity Pixels

The analysis software used has the pixel amplitude corresponding to a numerical scale set between 0-255. By selecting a certain interval from the 0-255 scale, we obtained values corresponding to a certain pixel type, present in the analyzed image. Thus, the 0-30 interval corresponds to low echogenic pixels (LEP), the 50-150 interval to medium echogenicity pixels, and the 200-255 interval to high echogenicity pixels.

2.4 Measurement of Low, Medium and High Echogenicity Pixels

SLEB was defined as a well delimited, subepidermal low echogenicity band (0-30), situated in the upper dermis, mainly present on photoexposed sites [6]. In order to establish SLEB, we evaluated the number of low echogenic pixels (0-30) situated in the upper dermis.

2.5 Quantification of LEPs/LEPi Ratio

LEP was determined at the dermal level, between the epidermis and the hypodermis. Additionally, the LEP area was divided into two other regions, differently quantified: superior LEP (LEPs), and inferior LEP (LEPi). The limit between the two regions was obtained by dividing the dermis into two equal parts, by drawing a parallel line to the epidermis echogenicity line. The LEPs/LEPi ratio was established. This ratio allows an appreciation of the density and integrity of the extracellular matrix, both from the papillary and lower dermis, which may vary according to age, cutaneous affections, UV exposure, therapy [7, 8].

3 Results

3.1 Assessment of Low, Medium and High Echogenicity Pixels

The number of hypoechogenic pixels shows a significant variation in case of the dorsal forearm and medial arm of the patients taken into study, as follows: hypoechogenic pixels significantly decrease on the dorsal forearm in the 20-40 age interval compared to the 4-20 interval (p= 0.038018, p<0.05) and increase significantly in the >60 age interval in comparison to the 41-60 interval (p= 0.00777, p<0.05); on the medial arm, hypoechogenic pixels increase significantly in the 41-60 age interval, compared to 20-40 interval (p= 0.018056, p<0.05).

Intermediate pixels display statistical significant values on the dorsal forearm and face: they increase in the 20-40 age interval and decrease in subjects belonging to the 40-60 interval (p<0.05).

Hyperechogenic pixels also display statistically significant variation on the three analyzed regions: on the dorsal forearm, high echogenic pixels increase significantly in the 20-40 interval of age, compared to the 4-20 interval (p=0.025154, p<0.05, and slightly decrease after the age of 40; on the medial arm, they decrease in the 40-60 age interval compared to the 20-40 age interval (p= 0.038523, p<0.05) and at facial site, high echogenic pixels increase in the 21-40 interval (p= 0.025405, p< 0.05) and decrease between 41-60 (p= 0.048694, p<0.05).

Thus, the number of hyperechogenic pixels may be considered a marker of intrinsic senescence. The highest amount of hypoechogenic pixels was identified at facial level, an intensely photoagressed site, whereas the highest amount of hyperechogenic pixels was found at the medial arm site, a less photoexposed area. High echogenic pixels (200-255) are poorly expressed in patients belonging to the 4-20 age interval, and much better expressed in the 20-40 age interval on all studied areas.

In the 4-20 age interval, on the dorsal forearm and medial arm, there is a predominance of hypoechogenic pixels (0-30), in comparison to intermediate echogenic pixels (50-100). An opposite correlation may be noticed at facial site, where intermediate echogenic pixels appear in greater amount than the low echogenic ones.

3.2 SLEB Assessment (LEP)

SLEB was identified in case of the subjects part of the 41-60 and >60 age intervals, and appeared especially on photoexposed sites (dorsal forearm, face) [9,10]

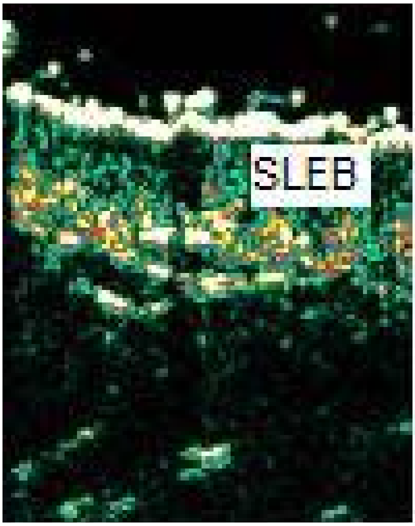


Fig. 2. SLEB present in the upper dermis at dorsal arm level

(see Figure 2). In some patients though, especially the younger ones, we were able to identify SLEB at medial arm level as well. On photo-aggressed sites, it may be noticed that the echogenicity of the upper dermis decreases with age.

3.3 The Assessment of LEPs/LEPi Ratio

The ultrasound study shows different echogenicity degrees for the upper (LEPs) and lower (LEPi) dermis.

For the upper dermis, the study revealed an increase of hypoechogenic pixels (0-30), in comparison to the lower dermis, for all 4 age intervals studied. The hypoechogenicity degree is higher on photoexposed sites, both for the upper and the lower dermis.

Evaluating the LEPs/LEPi ratio for all age intervals, a significant growth of the ratio value is noticed for the >60 age interval, in case of all three examined sites, due to a significant growth of hypoechogenic pixels in the upper dermis.

A significant decrease of hypoechogenic pixels for the 20-40 age interval was noticed, for all three examined sites, followed by an increase for the 40-60 and >60 age intervals.

For the 20-40 age interval, a significant increase, almost a double value was identified in case of the high echogenicity pixels (200-255), followed by a decrease in the following age intervals.

4 Discussion

High frequency ultrasound represents a noninvasive, “in vivo” method of histological study of the integument [11.12]. It allows a specific appreciation of the tissular echogenicity degree that varies with age, UV exposure, or different general or topical therapies. Even on artificial skin there were ecographic studies that provided high knowledge about the skin models, concluding that high frequency ultrasonography is a valuable non invasive method of diagnosis [13].

This method allows both a qualitative and quantitative appreciation of the cutaneous hydration degree, the degenerative changes of collagen and elastic fibres, the degree of collagen glycation [14,15] and other biochemical variations of the extracellular matrix.

The main source of dermal echogenicity is represented by collagen fibres, disposed in an organized manner. Collagen and elastin are key proteins in maintaining the cutaneous architecture and ensuring the biostructural qualities of the integument system [16,17].

During the aging process, integument collagen suffers a structural disorganization process as well as different biochemical changes [18, 19]. It is possible that, at facial site, on actinic aggressed integument, the decrease of collagen content and the changes of the fundamental substance may be counter balanced by a global rearrangement of the dermal collagen network and the accumulation of elastotic material. Biochemical and biostructural changes at the facial site are also possible, predisposing to the appearance of certain pathology according to age (actinic keratosis, carcinomas, malignant melanomas) [20].

The significant increase ($p<0.05$) of hypoechogenic pixels after the age of 40, both on photoexposed and photoprotected sites, is correlated with the degenerative changes which are typical for the aging process in general. Initially, elastic fibres from the papillary dermis are altered, leading to the appearance of fine wrinkles and venectasies associated to age, then, collagen fibres are altered, leading to the appearance of the deep wrinkles. At the same time, oxythalan fibres get disorganized, being responsible for the cutaneous fragility of the elderly [21]. Generally, we noticed that hypoechigenic pixels are more numerous in the upper dermis in elderly subjects on all sites studied.

SLEB is a specific parameter that varies in thickness and localization according to age and UV exposure. In young subjects, SLEB is present in the lower dermis and quantifies the degree of cutaneous hydration [22], since the extracellular matrix is rich in proteoglycans and hyaluronic acid [7, 23]. Hyaluronic acid binds uncovalently the proteoglycans, forming macromolecules that attract water, forming a true hydrating capsule. In elderly subjects, SLEB quantifies the elastosis process [24] and basophilic degenerescence of collagen, common aspects of the senescence process, but increased by UV [25]. According to literature, SLEB is better expressed on photoexposed sites in comparison to unexposed zones, for all age intervals. Thus, we may consider SLEB as a qualitative marker of the photoagression process.

Intemediate echogenic pixels increase significantly statistic ($p<0.05$) on photoexposed sites in the 20-40 age interval, and decrease after the age of 40. The dynamics of the repartition of intermediate echogenic pixels shows that UV rays initiate changes at biochemical and structural level, in case of the 20-40 age interval. The significant echogenicity variations in this “critical interval” of age, indicates the presence of intense structural processes that continue on to the next intervals of age, but in a much slower rhythm. We consider that the photoinduced cutaeneous pathology that may usually be evidenced after the age of 45, can be ameliorated through efficient protection measures applied until the age of 40 [26].

The number of hyperechogenic pixels (200-255) increase significantly ($p<0.05$) in the 20-40 age interval on photoexposed sites and decrease significantly on photoprotected areas in the 40-60 age interval. Their evolution is directly influenced by the UV exposure. According to Table III, the mean of hyperechogenic pixels is higher on photoprotected areas compared to the photoagressed ones for all intervals of age. Thus, we may consider hyperechogenic pixels as ultrasound markers of chronological aging.

LEPs/LEPi ratio shows a statistically significant increase ($p<0.05$) for the 20-40 and 40-60 age intervals on photoexposed sites, especially at facial level ($p= 0.000999$, $p<0.05$). On the medial arm, a progressive decrease is noticed till the age of 60, followed by a light increase in people >60 years. This aspect may be explained by the

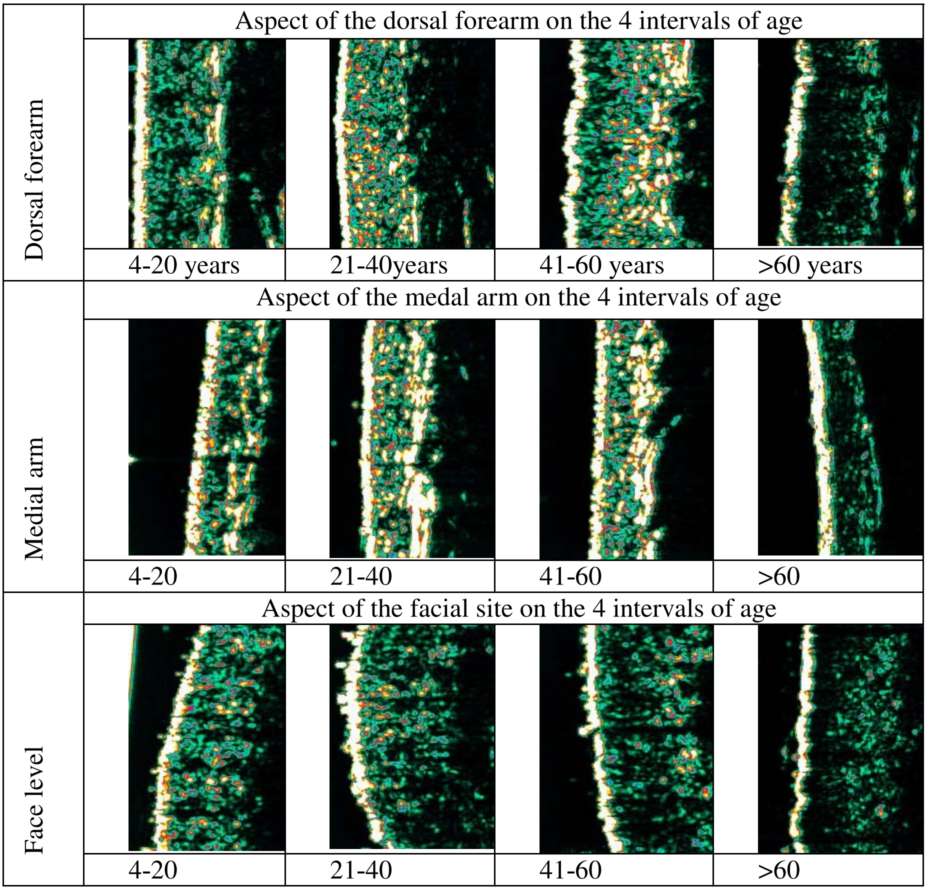

Dorsal forearm
Aspect of the dorsal forearm on the 4 intervals of age
4-20 years
21-40years
41-60 years
>60 years
Medial arm
Aspect of the medal arm on the 4 intervals of age
4-20
21-40
41-60
>60
Face level
Aspect of the facial site on the 4 intervals of age
4-20
21-40
41-60
>60

Fig. 3. Ultrasound aspect of the dorsal arm, medial arm and facial site on the 4 studied age

increase of hypoechogenic pixels in the upper dermis (see Figure 5). Unlike the upper dermis, in the lower dermis, an increase of echogenicity may be noticed with aging. (Table 4). The ratio between the echogenicity of the upper and lower dermis represents an objective marker of the photoaging process.

The variations of cutaneous echogenicity identify both structural and biochemical changes related to age, and the degree of photo-aggression. It is known that the alteration process of the elastic fibres begins around the age of 30. This process is accelerated by UV rays, affecting thus all cutaneous structures [27]. The next step is to implement a mathematical model of the aging skin for further skin analysis and for correctly evaluate and foresee the evolution of aging skin. The mathematical model that we suggest is a fractal one, considering that the aging process of the skin is developing on such a model. In general, the scientific world of today tries to accomplish different mathematical models for different processes (weather cast, physics, chemistry and so on). The medicine has very shy attempts (till now) to characterize any process via a mathematical model. It would be a very accomplishing fact to predict different medical processes, to appreciate the efficiency of a therapy using several parameters integrated in a model. There are some examples in which the authors try to explain a mathematical model for causes of mortality, or, relating the sun exposure to a future skin cancer. [28], [29]. Our collective is developing a mathematical model to characterize the aging of the skin with variables considered from high frequency ultrasonography regarding the echogenity of the dermis.

5 Conclusion

High frequency ultrasound is a modern method that offers specific parameters which allow the noninvasive appreciation of both physiological and pathological aspects of the integumentary system.

The thicknesses of the integument, SLEB, as well as the dermal echogenicity are parameters that evaluate, with high accuracy the cutaneous senescence level at a microscopical level. The ratio between the echogenicity of the upper and lower dermis represents an objective marker of the aging process. Currently a mathematical model to characterize the aging of the skin is under study.

Acknowledgments. The authors gratefully acknowledge Cortex Technology for allowing the use of the ultrasound equipment and for their assistance with the application of the Dermascan to this project.

The study is part of a research grant SERENO number 2624 heading should be treated as a 3rd level heading and should not be assigned a number.

References

1. Gniadecka, M., Jemec, G.B.E.: Quantitative evaluation of chronological aging and photoageing in vivo: studies on skin echogenicity and thickness. British Journal of Dermatology D92, 138:00-00 (1998)
2. Monika-Hildegard Schmid-Wendtner, M.D., Walter Burgdorf, M.D.: Ultrasound Scanning in Dermatology. Arch Dermatol 141, 217–224 (2005)

3. Lasagni, C., Seidenari, S.: Echographic assessment of age dependant variations of skin thickness: a study on 162 subjects. Skin Res Technol 1, 81–85 (1995)
4. Tsukhara, K., Takema, Y., Moriwaki, S., Fujimura, T., Kitahara, T., Immokava, G.: Age related alterations of echogenicity in Japanese skin. Dermatology 200, 303–307 (2000)
5. Seidenari, S., Pagnoni, A., Dinando, A., Giannetti, A.: Echographic Evaluation with Image Analysis of Normal Skin: Variations according to age and sex. Skin Pharmacol 7, 201–209 (1994)
6. Gniadecka, M., Gniadecki, R., Serup, J., Søndergaard, J.: Ultrasound Structure and Digital Image Analysis of the Subepidermal Low Echogenic Band in Aged Human Skin: Diurnal Changes and Interindividual Variability. Journal of Investigative Dermatology 102, 362–365 (1994), doi:10.1111/1523-1747.ep12371796.
7. Gniadecka, M.: Effects of aging on dermal echogenicity. Skin Research and Technology 7, 204–207 (2001)
8. Brancheta, M.C., Boisnicb, S., Francesc, C., Roberta, A.M.: Skin Thickness Changes in Normal Aging Skin. International Journal of Experimental, Clinical Behavioural, Regenerative and Technological Gerontology 36(1), 28–35 (1990)
9. Pellacani, G., Seidenari, S.: Variations in facial skin thickness and Echogenicity with site and age. Acta Dermatologica Venerologica 79, 366–369 (1999)
10. Lacarrubba, F., Tedeschi, A., Nardone, B., Micali, G.: Mesotherapy for skin rejuvenation: assessment of the subepidermal low-echogenic band by ultrasound evaluation with cross-sectional B-mode scanning. In: Dermatologic Therapy, November/December 2008, vol. 21(3), pp. S1–S5(1). Blackwell Publishing, Dermatologic Therapy (2008)
11. Jemec, G.B.E., Gniadecka, M., Ulrich, J.: Ultrasound in dermatology. Part I. High frequency ultrasound. EJ D. European journal of dermatology ISSN 1167-1122 10(6), 492–497 (2000) (40 ref.)
12. Szymańska, E., Nowicki, A., Mlosek, K., Litniewski, J., Lewandowski, M., Secomski, W., Tymkiewicz, R.: Skin imaging with high frequency ultrasound — preliminary results European Journal of Ultrasound, September 2000, vol. 12(1), pp. 9–16 (2000)
13. Iwamoto, T., Saijo, Y., Hozumi, N., Kobayashi, K., Okada, N., Tanaka, A., Yoshizawa, M.: High frequency ultrasound characterization of artificial skin. In: Conf Proc IEEE Eng Med Biol Soc. 2008, pp. 2185–2188 (2008)
14. Kasper, M., Funk, R.H.W.: Age-related changes in cells and tissues due to advanced glycation end products (AGEs). Archives of Gerontology and Geriatrics 32, 233–243 (2001)
15. Reiser, K.M.: Nonenzymatic glycation of collagen in aging & diabetes. In: Proc. Soc.Exp.Biol.Med., vol. 218, pp. 23–37 (1998)
16. Uitto, J.: The role of elastin and collagen in cutaneous aging: intrinsic aging versus photoexposure. J Drugs Dermatol, Feburary 7(2 Suppl) :s12-6 (2008)
17. Seidenari, S., Giusti, G., Bertoni, L., Magnoni, C., Pellacani, G.: Thickness and echogenicity of the skin in children as assessed by 20-MHZ ultrasound. Dermatology 201, 218–222 (2000)
18. Fligiel, S.E.G., Varani, J., Datta, S.C., Kang, S., Fisher, G.J., Voorhees, J.J.: Collagen Degradation in Aged/Photodamaged Skin In Vivo and After Exposure to Matrix Metalloproteinase-1 In Vitro. Journal of Investigative Dermatology 120, 842–848 (2003)
19. Nishimori, Y., Edwards, C., Pearse, A., Matsumoto, K., Kawai, M., Marks, R.: Degenerative Alterations of Dermal Collagen Fiber Bundles in Photodamaged Human Skin and UV-Irradiated Hairless Mouse Skin: Possible Effect on Decreasing Skin Mechanical Properties and Appearance of Wrinkles. Journal of Investigative Dermatology 117, 1458–1463 (2001)

20. Rexbye, H., Petersen, I., Johansens, M., Klitkou, L., Jeune, B., Christensen, K.: Influence of environmental factors on facial ageing. Age and Ageing 35(2), 110–115 (2006)
21. Lee, J.Y., Kim, Y.K., Sea, J.Y., Choi, C.W., Hwang, J.S., Lee, B.G., Chang, I.S., Chung, J.H.: Loss of elastic fibers causes skin wrinkles in sun-damaged human skin. Journal of dermatological science ISSN 0923-1811 50(2), 99–107 (2008)
22. Eisenbeiss, C., Welzel, J., Eichler, W., Klotz, K.: Influence of body water distribution on skin thickness: measurements using high-frequency ultrasound. Br J Dermatol 144, 947–951 (2001)
23. Sator, G., Schmidt, J.B., Hönigsmann, H.: objective assessment of photoageing effects using high-frequency ultrasound in PUVA-treated psoriasis patients. British Journal of Dermatology 147(2), 291–298
24. Diana, C., Calderone, M.D.: The clinical spectrum of actinic elastosis. Journal of the American Academy of Dermatology - 32(6) (June 1995)
25. Yaar, M., Glichrest, B.A.: Skin aging: postulated mechanisms and consequent changes in structure and function. Clin Geriart Med 17, 617–630 (2001)
26. Stephan Lautenschlager, Hans Christian Wulf dsc b, Prof Mark R Pittelkow M D c. Photoprotection. The Lancet, Volume 370, Issue 9586, Pages 528 - 537, 11, Original Text (August 2007)
27. Fisher, G.J., Kang, S., Varani, J., et al.: Mechanisms of photoaging and chronological skin aging. Arch Dermatol 2002 138, 1462–1470 (2002)
28. Fears, T.R., Andmarvin, J.S., Schneiderman, A.: Ultraviolet effects on the incidence of skin cancer among whites in the united states. Field Studies and Statistics Program and the Demography Section, Biometry Branch, National Cancer Institute Bethesda, MD 20014
29. Stephen Brown., K., Math., B., Forbes, W.F.: PhD, DSc2; A Mathematical Model of Aging Processes. II1; Dept. of Statistics, Faculty of Mathematics, Univ. of Waterloo Waterloo, Ont. N2L 3G1
30. DermaScan, C.: Cortex Technology, Denmark, `http://www.cortex.dk/dermascan_c.htm`

Self-similar Hierarchical Regular Lattices

Carlo Cattani[1] and Ettore Laserra[2]

[1] diFarma
[2] ⟨ DMI ⟩,
University of Salerno,
Via Ponte Don Melillo, 84084 Fisciano (SA) Italy
ccattani@unisa.it, elaserra@unisa.it

Abstract. This paper deals with the topological-metric structure of a network made by a family of self-similar hierarchical regular lattices. We derive the basic properties and give a suitable definition of self-similarity on lattices. This concept of self-similarity is shown on some classical (omothety) and more recent models (Sierpinski tesselations and Husimi cacti). Both the metric and the geometric properties of the lattice will be intrinsically defined.

1 Introduction

In recent years many physical problems such as spin glasses, renormalization groups, coherent transport, hyperbranched macromolecules and dendrimers, molecular crystals, aggregates and diamond lattices, complex networks [1,2,3,7,8,12,14,20,21,22,23], lead to the analysis of lattices, or agglomerated patterns [4,11]. It has been already shown that the application of the simplicial calculus or calculus on regular lattices might lead to simplified theory such as discretized gravitational Einstein theory, so-called Regge calculus [6,9,13,15].

In this paper we will study some regular lattices made on the transformation group of a simplicial complex. Starting from a simplex it will defined the group of transformation so that the intrinsic (affine) metric is scale invariant. It is shown that some known lattices can be obtained by suitable maps in the defined group.

2 Preliminary Definitions

The euclidean m-simplex σ^m, of independent vertices $V_0, V_1, ..., V_m$, is defined [10,16,19] as the subset of $\mathbb{R}^n$:

$$\sigma^m \overset{\text{def}}{=} \left\{ P \in \mathbb{R}^n \mid \quad P = \sum_{i=0}^{m} \lambda^i V_i \text{ with } \sum_{i=0}^{m} \lambda^i = 1 \ , \ 0 \leq \lambda^i \leq 1 \right\} .$$

We can also say that $V_0, V_1, ..., V_m$ is a *barycentric basis* for all $P \in \sigma^m$ so that λ^i are the *barycentric coordinates* of P in σ^m. When $0 < \lambda^i < 1$ we have the inner simplex $\overset{\circ}{\sigma}{}^m$.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 225–240, 2010.

Let us denote with $[\sigma^m] = [V_0, V_1, ..., V_m]$ the set of points which form the *skeleton* of σ^m, the p-face of σ^m, with $p \leq m$, is any simplex σ^p such that $[\sigma^p] \cap [\sigma^m] \neq \emptyset$, and we write $\sigma^p \preceq \sigma^m$.

The number of p-faces of σ^m are $\begin{pmatrix} m+1 \\ p+1 \end{pmatrix}$.

The m-dimensional *simplicial complex* Σ^m is defined as the finite set of p simplices $(p \leq m)$ such that:

1. $\forall \sigma^k \in \Sigma^m$, if $\sigma^h \preceq \sigma^k$ then $\sigma^h \in \Sigma^m$,
2. $\forall \sigma^k, \sigma^h \in \Sigma^m$ then either $[\sigma^h] \cap [\sigma^k] = \emptyset$ or $[\sigma^h] \cap [\sigma^k] = [\sigma^j]$ with $\sigma^j \in \Sigma^m$.

For a given $\sigma^p \in \Sigma^m$, the *star* $St(\sigma^p)$ *of* σ^p is $St(\sigma^p) \overset{\text{def}}{=} \{\sigma^q \in \Sigma^m \,|\, \sigma^p \preceq \sigma^q\}$.

The set of points P such that $P \in \sigma^p$, $p \leq m$ and $\sigma^p \in \Sigma^m$ is the geometric support of Σ^m also called m-polyedron $\mathcal{M}^m$. The p-skeleton of Σ^m is $[\Sigma^m]^p \overset{\text{def}}{=} [\sigma^p]$ $, \forall \sigma^p \in \Sigma^m$. The *boundary* $\partial \Sigma^m$ of Σ^m is the complex Σ^{m-1} such that each $\sigma^{m-1} \in \Sigma^{m-1}$ is face of only one m-simplex of Σ^m.

2.1 Orientation

Each skeleton $[V_{i_0},\ V_{i_1},\ \ldots,\ V_{i_m}]$ of a p-simplex is considered *oriented* if the class (even or odd) permutations of indices $[i_0,\ i_1,\ \ldots, i_m]$ with respect to the fundamental ordered set $i_0 < i_1 < \ldots < i_m$ is given. The orientation of the skeleton implies the orientation of the corresponding simplex.

Let us consider a few examples:

a) $m = 2$: A two simplex with vertices V_0, V_1, V_2 corresponds to the oriented simplices: $\{V_0, V_1, V_2\}$ o $\{V_0, V_2, V_1\}$.
b) $m = 3$: For a 3–simplex $\{V_0, V_1, V_2, V_3\}$ we have two classes of simplices: $\{V_0, V_1, V_2, V_3\}$, $\{V_0, V_2, V_3, V_1\}$, $\{V_0, V_3, V_1, V_2\}$, obtained by even permutation of indices and the odd class of permutations $\{V_0, V_2, V_1, V_3\}$, $\{V_0, V_1, V_3, V_2\}$, $\{V_0, V_3, V_2, V_1\}$.

The orientation of the simplex implies a natural orientation of the boundary as follows. Given an ordered set of vertices $[V_0, \ldots, V_m]$, the k-th face of σ^m, denoted by $\sigma^m_{,k}$ $(0 \leq k \leq m)$, is naturally oriented as

$$\sigma^m_{,k} \overset{def}{=} (-1)^k [V_0, \ldots, \hat{V}_k, \ldots, V_m] \qquad 0 \leq k \leq m \tag{1}$$

where the symbol " ˆ " stands for missing vertex V_k.

The *oriented boundary* $\partial \sigma^m$ of σ^m is the set of faces naturally oriented

$$\partial \sigma^m = \sum_{i=0}^{m} \sigma^m_{,i} = \sum_{i=0}^{m} (-1)^i [V_0, \ldots, \hat{V}_i, \ldots, V_m] \tag{2}$$

There easily follows the known principle [15] that the boundary of the boundary vanishes

$$\partial\partial\sigma^m = \sum_{i=0}^{m} (-1)^{i+j} [V_0, \ldots, \hat{V}_i, \ldots, \hat{V}_j, \ldots, V_m] = 0 \ . \tag{3}$$

The Σ^m complex is oriented if it possible to give an orientation to each simplex in a such way that adjacent simplices, such that the intersection of corresponding boundaries is a non empty set, are counter-oriented.

3 Barycentric Coordinates and Barycentric Bases

In each simplex it is possible to define the *barycentric basis* as follows: given the m-simplex σ^m with vertices $V_0, ..., V_m$,the barycentric basis is the set of $(m+1)$ vectors

$$\mathbf{e}_i \overset{\text{def}}{=} V_i - \mathcal{G}^m \tag{4}$$

based on the *barycenter*

$$\mathcal{G}^m \overset{\text{def}}{=} \mathcal{G}(\sigma^m) = \sum_{i=0}^{m} \frac{1}{m+1} V_i \ .$$

These vectors are linearly dependent, since according their definition it is:

$$\sum_{i=0}^{m} \mathbf{e}_i = \mathbf{0} \quad . \tag{5}$$

Each point $P \in \sigma^m$ can be characterized by a set of *barycentric coordinates* $(\lambda^0, ...\lambda^m)$ such that

$$0 \leq \lambda^i \leq 1 \quad , \quad \sum_{i=0}^{m} \lambda^i = 1 \quad , \ i = 0, \ldots, m \tag{6}$$

and $P - \mathcal{G}^m = \sum_{i=0}^{m} \lambda^i \mathbf{e}_i \overset{(4),(6)}{\Longrightarrow} P = \sum_{i=0}^{m} \lambda^i V_i$. Therefore each point of σ^m can be formally expressed as a linear combination of the skeleton $[\sigma^m]$.

Let P and Q be a couple of points in σ^m, with coordinates $(\lambda^0_P, ...\lambda^m_P)$ and $(\lambda^0_Q, ...\lambda^m_Q)$ respectively, it is

$$\mathbf{v} = Q - P = \sum_{i=0}^{m} v^i \mathbf{e}_i \quad \text{e} \quad \sum_{i=0}^{m} v^i = 0 \tag{7}$$

with $v^i = \lambda^i_Q - \lambda^i_P$.

3.1 Dual Space

The dual space is defined as the linear map of the vector space E into $\mathbb{R}$ as:

$$< \mathbf{e}^i, \mathbf{e}_k >= \tilde{\delta}^i_k \tag{8}$$

with [7]

$$\tilde{\delta}_k^i \overset{\text{def}}{=} \delta_k{}^i - \frac{1}{m+1} = \begin{cases} = -\dfrac{1}{m+1} & , i \neq k \\ = +\dfrac{m}{m+1} & , i = k \end{cases} \tag{9}$$

$\delta_k{}^i$ being the Kroneker symbol. According to the definition (9) it is

$$\sum_{i=0}^{m} \tilde{\delta}_k^i = \sum_{k=0}^{m} \tilde{\delta}_k^i = 0 \tag{10}$$

Examples

a) $m = 1$: We have:

$$< \mathbf{e}^0, \mathbf{e}_0 >=< \mathbf{e}^1, \mathbf{e}_1 >= +\frac{1}{2}$$

$$< \mathbf{e}^0, \mathbf{e}_1 >=< \mathbf{e}^1, \mathbf{e}_0 >= -\frac{1}{2}$$

so that

$$\sum_{i=0}^{1} < \mathbf{e}^i, \mathbf{e}_0 >= \sum_{i=0}^{1} < \mathbf{e}^i, \mathbf{e}_0 >= 0 \Rightarrow < \sum_i \mathbf{e}^i, \mathbf{e}_i >= 0$$

which implies

$$\sum_{i=0}^{1} \mathbf{e}^i = \mathbf{0} \quad . \tag{11}$$

b) $m = 2$: With $m = 2$ we have

$$< \mathbf{e}^0, \mathbf{e}_0 >=< \mathbf{e}^1, \mathbf{e}_1 >=< \mathbf{e}^2, \mathbf{e}_2 >= +\frac{2}{3}$$

$$< \mathbf{e}^0, \mathbf{e}_1 >=< \mathbf{e}^0, \mathbf{e}_2 >=< \mathbf{e}^1, \mathbf{e}_2 >= -\frac{1}{3} \ ,$$

that is

$$\sum_{i=0}^{2} < \mathbf{e}^i, \mathbf{e}_0 >= \sum_{i=0}^{2} < \mathbf{e}^i, \mathbf{e}_1 >= \sum_{i=0}^{2} < \mathbf{e}^i, \mathbf{e}_2 >= 0$$

and $< \sum_i \mathbf{e}^i, \mathbf{e}_k >= 0$ from where $\displaystyle\sum_{i=0}^{2} \mathbf{e}^i = \mathbf{0}$.

Analogously, it can be shown that in general it is

$$\sum_{i=0}^{m} \mathbf{e}^i = 0 \quad . \tag{12}$$

Since $\mathbf{e}^i$ are linear functions it is $<\mathbf{e}^i, \mathbf{0}>= 0$, with

$$0 =< \mathbf{e}^i, 0 >=< \mathbf{e}^i, \sum_{k=0}^{m} \mathbf{e}_k >= \sum_{k=0}^{m} \tilde{\delta}^i_k \overset{(10)}{=} 0 \ .$$

Let us now denote with $E_{(i)}$ the vector space to which the $(m-1)$–face $\sigma^m_{,i}$ of σ^m belongs. We can say that this face is somehow opposite to the vertex V_i, in the sense that $\mathbf{e}^i$ is orthogonal, according to (10), to any vector of $E_{(i)}$

$$\forall \mathbf{v} \in E_{(i)} \qquad < \mathbf{e}^i, \mathbf{v} >= 0 \quad . \tag{13}$$

In fact, for a fixed k such that $k \neq i$, let say $k = 0$ it is:

$$\mathbf{v} = \sum_k v^k (V_k - V_0) = \sum_k v^k (\mathbf{e}_k - \mathbf{e}_0) \qquad k \neq i \quad ,$$

from where

$$< \mathbf{e}^i, \mathbf{v} >= \sum_k v^k [< \mathbf{e}^i, \mathbf{e}_k > - < \mathbf{e}^i, \mathbf{e}_0 >] = \sum_{k \neq i} v^k \left[-\frac{1}{m+1} + \frac{1}{m+1} \right] = 0 \ .$$

4 Affine Metrics

The metric tensor in σ^m is defined as [8]

$$\boxed{\tilde{g}_{ij} \overset{\text{def}}{=} -\frac{1}{2} \tilde{\delta}^h_i \tilde{\delta}^k_j \ell^2_{hk}} \quad , \quad (i, j, h, k = 0, 1, \ldots, m) \tag{14}$$

being $\ell^2_{hk} \overset{\text{def}}{=} (V_k - V_h)^2 = (\mathbf{e}_k - \mathbf{e}_h)^2$.

If we consider the set $\{\ell^2_{ij}\}^m_{i,j=0} = \{\ell^2_{00},\ \ell^2_{01}, \ldots,\ \ell^2_{(m-1)m},\ \ell^2_{mm}\}$ as independent variables we have:

$$\frac{\partial \tilde{g}_{ij}}{\partial \ell^2_{rs}} = -\frac{1}{2} \tilde{\delta}^h_i \tilde{\delta}^k_j \frac{\partial l^2_{hk}}{\partial \ell^2_{rs}} = -\frac{1}{2} \tilde{\delta}^r_i \tilde{\delta}^s_j - \frac{1}{2} \tilde{\delta}^r_j \tilde{\delta}^s_i = -\frac{1}{2} \tilde{\delta}^r_{(i} \tilde{\delta}^s_{j)} \quad . \tag{15}$$

Given a tensor $\mathbf{A}^{ij}$ it is

$$< \tilde{g}, \mathbf{A} >= \sum_{ij} \mathbf{A}^{ij} \tilde{g}_{ij} = -\frac{1}{2} \sum_{\substack{ij \\ hk}} (\mathbf{A}^{ij} \tilde{\delta}^h_i \tilde{\delta}^k_j) l^2_{hk} = -\frac{1}{2} \sum_{hk} \mathbf{A}^{hk} l^2_{hk} \quad . \tag{16}$$

The inverse metrics can be defined as follows. According to (10) it is

$$\begin{vmatrix} \tilde{g}_{00} & \cdots & \tilde{g}_{0m} \\ \vdots & \ddots & \vdots \\ \tilde{g}_{m0} & \cdots & \tilde{g}_{mm} \end{vmatrix} = 0 \quad , \tag{17}$$

however, since

$$\begin{vmatrix} \sum_{i=0}^{m}\tilde{g}_{i0} & \cdots & \sum_{i=0}^{m}\tilde{g}_{im} \\ \vdots & \ddots & \vdots \\ \tilde{g}_{m0} & \cdots & \tilde{g}_{mm} \end{vmatrix} = 0 \ , \tag{18}$$

we cannot have (by a direct method) the inverse matrix being $\tilde{g}$ a singular matrix.

However, to any $\mathbf{v}$ di E, corresponds a linear form $H = \sum_{i=o}^{m} H_i \mathbf{e}^i$ di E^*, with

$$\begin{cases} H_i & = & \tilde{g}_{ik} v^k \\ \sum_{k=0}^{m} v^k & = & 0 \end{cases} \tag{19}$$

The same matrix can be obtained by solving the following equation

$$\begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & \tilde{g}_{00} & \cdots & \tilde{g}_{0m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \tilde{g}_{m0} & \cdots & \tilde{g}_{mm} \end{pmatrix} \begin{pmatrix} 0 \\ v^0 \\ \vdots \\ v^m \end{pmatrix} = \begin{pmatrix} 0 \\ H_0 \\ \vdots \\ H_m \end{pmatrix} \quad . \tag{20}$$

System (19) admits a unique solution under the condition:

$$\begin{vmatrix} 0 & 1 & \cdots & 1 \\ 1 & \tilde{g}_{00} & \cdots & \tilde{g}_{0m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \tilde{g}_{m0} & \cdots & \tilde{g}_{mm} \end{vmatrix} = \left(-\frac{1}{2}\right)^m \begin{vmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & \ell_{01}^2 & \cdots & \ell_{0m}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \ell_{m0}^2 & \cdots & \cdots & 0 \end{vmatrix} \neq 0 \quad , \tag{21}$$

which is a consequence of the independence of the vertices.

Thus we can define $\tilde{g}_{ij}$ in (21) by

$$\begin{cases} \sum_{i} \tilde{g}_{ih} \tilde{g}^{ik} = \tilde{\delta}_h^k \\ \sum_{i=0}^{m} \tilde{g}^{ik} = \sum_{k=0}^{m} \tilde{g}^{ik} = 0 \end{cases} \tag{22}$$

or, equivalently:

$$\begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & \tilde{g}_{00} & \cdots & \tilde{g}_{0m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \tilde{g}_{m0} & \cdots & \tilde{g}_{mm} \end{pmatrix} \begin{pmatrix} 0 & \frac{1}{m+1} & \cdots & \frac{1}{m+1} \\ \frac{1}{m+1} & \tilde{g}^{00} & \cdots & \tilde{g}^{0m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{m+1} & \tilde{g}^{m0} & \cdots & \tilde{g}^{mm} \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \tag{23}$$

It can be seen that, with this definition, it is

$$\sum_{hk} \tilde{g}^{ih}\tilde{g}^{jk}g_{hk} = \sum_{h} \tilde{g}^{ih}\tilde{\delta}^{h}_{j} = \tilde{g}^{ij} \ , \quad \sum_{h} \tilde{g}^{ih}\tilde{g}_{ih} = \tilde{\delta}^{h}_{i} \ .$$

5 Volume of an m-Simplex

In each simplex σ^m with vertices $V_0, \ldots, V_m$ we can define:

$$\omega^{j_0,j_1,\ldots,j_m} = (m!)^{-1}\epsilon^{j_1,\ldots,j_m} \tag{24}$$

where

$$\epsilon^{j_1,\ldots,j_m} = \begin{cases} 1 \\ -1 \end{cases}$$

with respect to the even or odd class of permutation of $(j_0, j_1, \ldots, j_m)$ with respect to the fundamental ordering $\{0, 1, \ldots, m\}$.

The volume of the simplex is defined by

$$\Omega(\sigma^m) \overset{\text{def}}{=} \pm \sum_{\substack{j_1,\ldots,j_m \\ k_1,\ldots,k_m}} \frac{1}{m!}\omega^{j_1,\ldots,j_m}\omega^{k_1,\ldots,k_m} \prod_{a=1}^{m} \tilde{g}_{j_a k_a} \quad , \tag{25}$$

where ± depends on the ortientation of the simplex.

According to (16),(24) the previous equation is equivalent to

$$\Omega(\sigma^m) = \pm \left(-\frac{1}{2}\right)^m \left(\frac{1}{m!}\right)^3 \sum_{\substack{j_1\cdots j_m \\ k_1\cdots k_m}} \tilde{\epsilon}^{j_1,\ldots,j_m}\tilde{\epsilon}^{k_1,\ldots,k_m} \prod_{a=1}^{m} \ell^2_{j_a k_a} \quad . \tag{26}$$

Moreover, the determinant of the matrix

$$\begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & \ell^2_{0m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \ell^2_{m0} & \cdots & 0 \end{pmatrix} \tag{27}$$

is

$$\frac{1}{m!}\epsilon^{j_1,\ldots,j_m}\epsilon^{k_1,\ldots,k_m} \prod_{a=1}^{m} \ell^2_{j_a k_a}$$

so that

$$\Omega(\sigma^m) = \mp \left(-\frac{1}{2}\right)^m \left(\frac{1}{m!}\right)^2 \begin{vmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & \ell^2_{0m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \ell^2_{m0} & \cdots & 0 \end{vmatrix} \tag{28}$$

We can show that the inverse metrics can be explicitly defined by the volume of the simplex that is

Theorem 1. *The inverse metrics is explicitly defined by*

$$\boxed{\tilde{g}^{ij} = (\delta^{ij} - 1)\frac{\partial \log \varOmega^2(\sigma^m)}{\partial \ell^2_{ij}} + \delta^{ij}\frac{\varOmega(\sigma^m_{,i})^4}{m^2\varOmega^2(\sigma^m)}} \quad . \tag{29}$$

Proof: From (23) it is

$$\tilde{g}^{ij} = \frac{\triangle_{ij}}{\varGamma}$$

where $\varGamma$ is the determinant of the matrix

$$\begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & \tilde{g}_{00} & \cdots & \tilde{g}_{0m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \tilde{g}_{m0} & \cdots & \tilde{g}_{mm} \end{pmatrix} \tag{30}$$

and $\triangle_{ij}$ is the co-factor of $\tilde{g}_{ij}$.

From

$$\varOmega^2(\sigma^m) = \left(\frac{1}{m!}\right)^2 \varGamma$$

by assuming $\{\ell^2_{ij}\}^m_{i,j=0}$ as independent variables we get

$$\frac{\partial \varOmega^2(\sigma^m)}{\partial \ell^2_{ij}} = \mp \left(-\frac{1}{2}\right)^m \left(\frac{1}{m!}\right)^2 2 \begin{vmatrix} 0 & 1 & \cdots & \cdots & 1 \\ 1 & 0 & \cdots & \cdots & \ell^2_{0m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \ell^2_{m0} & \cdots & \cdots & 0 \end{vmatrix} \tag{31}$$

$$= \mp \left(-\frac{1}{2}\right)^{m-1} \left(\frac{1}{m!}\right)^2 (-1)^{i+j} \begin{vmatrix} 0 & 1 & \cdots & \cdots & 1 \\ 1 & 0 & \cdots & \cdots & l^2_{0m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \ell^2_{i-1,0} & \cdots & \cdots & \ell^2_{i-1,m} \\ 1 & \ell^2_{i+1,0} & \cdots & \cdots & \ell^2_{i+1,m} \\ \cdots & \cdots & \cdots & \cdots\cdots & \\ 1 & \ell^2_{m0} & \cdots & \cdots & 0 \end{vmatrix} = \left(\frac{1}{m!}\right)^2 \triangle_{ij}$$

so that

$$\tilde{g}^{ij} = -\frac{1}{\varOmega^2(\sigma^m)}\frac{\partial \varOmega^2(\sigma^m)}{\partial \ell^2_{ij}} = -\frac{\partial \log \varOmega^2(\sigma^m)}{\partial \ell^2_{ij}}$$

follows.

When $i = j$ it is

$$\triangle_{ii} = (-1)^{2i} \begin{vmatrix} 0 & 1 & \cdots & \cdots & 1 \\ 1 & \cdots & \tilde{g}_{0,i-1} & \tilde{g}_{0,i+1} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \cdots & \tilde{g}_{i-1,i-1} & \tilde{g}_{i-1,i+1} & \cdots \\ 1 & \cdots & \tilde{g}_{i+1,i-1} & \tilde{g}_{i+1,i+1} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{vmatrix} = (m-1)!^2 \pm \varOmega^2(\sigma^m_{,i})$$

being $\Omega(\sigma^m_{,i})$ the volume of the $(m-1)$–face of σ in front of the vertex V_i.

Thus we have

$$\tilde{g}^{ii} = \frac{\mp(m-1)!^2 \Omega^2(\sigma^m_{,i})}{\mp(m)!^2 \Omega^2(\sigma^m)} = \frac{\Omega^4(\sigma^m_{,i})}{m^2 \Omega^2(\sigma^m)} \quad .$$

□

Let h_i be the distance of the vertex V_i by the opposite face $w\sigma^m_{,i}$:

$$h_i = dist(V_i, \sigma^m_{,i})$$

the volume $\Omega(\sigma^m)$ of σ^m can be computed as

$$\Omega(\sigma^m) = \frac{1}{m} h_i \Omega(\sigma^m_{,i}) \qquad \forall i \in \{0, \dots, m\} \quad ,$$

so that the inverse metrics for $i = j$ reduces to

$$\tilde{g}^{ii} = \frac{1}{h_i^2} \quad . \tag{32}$$

Let us define with $\mathbf{n}^i$ the normal vector to the face $\sigma^m_{,i}$:

$$\mathbf{n}^i \overset{\text{def}}{=} h_i \mathbf{e}^i \ . \tag{33}$$

The covariant components of this vector are

$$n^i_j = h_i \tilde{\delta}^i_j \quad . \tag{34}$$

It is also

$$n^{ik} = h_i \tilde{g}^{ki} \quad ; \tag{35}$$

in fact,

$$n^{ik} = \sum_j \tilde{g}^{kj} n^i_j = h_i \sum_j \tilde{\delta}^i_j \tilde{g}^{kj} = h_i \tilde{g}^{ki} = -\frac{m\,\Omega(\sigma^m)}{\Omega(\sigma^m_{,i})} \frac{\partial \log \Omega^2(\sigma^m)}{\partial \ell^2_{ki}} \quad . \tag{36}$$

So that we can write:

$$\mathbf{n}^i = h_i \sum_j \tilde{\delta}^i_j \mathbf{e}^j = h_i \sum_j \tilde{g}^{ki} \mathbf{e}_j \quad , \tag{37}$$

or, by taking into account (29),

$$\begin{aligned} \mathbf{n}^i &= \frac{m\,\Omega(\sigma^m)}{(\sigma^m_{,i})} \sum_j \tilde{\delta}^i_j \mathbf{e}^j \\ &= \frac{(\sigma^m_{,i})}{m\,\Omega(\sigma^m)} \mathbf{e}_i \frac{m\,\Omega(\sigma^m)}{(\sigma^m_{,i})} - \frac{1}{\Omega^2(\sigma^m)} \sum_{j \neq i} \frac{\partial \Omega^2(\sigma^m)}{\partial \ell^2_{ij}} \mathbf{e}_j \quad . \end{aligned} \tag{38}$$

It can be easily seen that $< \mathbf{n}^i, \mathbf{n}^i >= h^2_{(i)} < \mathbf{e}^i, \mathbf{e}^i >= \frac{1}{\tilde{g}_{ii}} \tilde{g}_{ii} = 1$.

Moreover, the boundary theorem holds

Theorem 2. *Let $\sigma^m_{,i}$ the face opposite to the vertex V_i and $\mathbf{n}^i$ the normal vector, it is:*

$$\sum_{i=0}^{m} \Omega(\sigma^m_{,i})\mathbf{n}^{(i)} = 0 \ . \tag{39}$$

Proof: From (33) we have

$$\sum_{i=0}^{m} \Omega(\sigma^m_{,i})\mathbf{n}^{(i)} = \sum_{i=0}^{m} \Omega(\sigma^m_{,i}) \frac{m\ \Omega(\sigma^m)}{\Omega(\sigma^m_{,i})}\mathbf{e}^i = m\ \Omega(\sigma^m) \sum_{i=0}^{m} e^i = 0$$

and

$$\sum_{i=0}^{m} \Omega(\sigma^m_{,i})n_j = \sum_{i=0}^{m} \Omega(\sigma^m_{,i}) \frac{m\Omega(\sigma^m)}{\Omega(\sigma^m_{,i})}\tilde{\delta}^i_j = m\ \Omega(\sigma^m) \sum_{i=0}^{m} \tilde{\delta}^i_j = 0 \ .$$

□

6 Simplicial Inequalities

This section deals with some inequalities on simplices.

a) $m = 1$: For each 1-simplex with vertices V_i, V_j, the condition

$$\ell^2_{ij} =< V_j - V_i, V_j - V_i > > 0 \tag{40}$$

holds.

b) $m = 2$: For a 2-simplex σ^2 with vertices V_i, V_j, V_k, equation ((40)) must be fulfilled together with $\Omega^2(\sigma^2) > 0$, where $\Omega(\sigma^2)$ can be expressed in terms of edge lengths as

$$\Omega^2(\sigma^2) = \rho(\rho - \ell_{ij})(\rho - \ell_{jk})(\rho - \ell_{ik}) \tag{41}$$

with $\rho = \frac{1}{2}(\ell_{ij} + \ell_{jk} + \ell_{ik})$.

There follows,

$$\Omega^2(\sigma^2) = \left(\frac{1}{2!}\right)^2 \begin{vmatrix} \ell^2_{ij} & c_{i,jk} \\ c_{i,kj} & \ell^2_{ik} \end{vmatrix} > 0 \tag{42}$$

with $c_{i,jk} = \frac{1}{2}(\ell^2_{ij} + \ell^2_{ik} - \ell^2_{jk}) = c_{i,kj}$. Thus we have

$$\ell^2_{ij}\ell^2_{ik} - c^2_{i,kj} > 0 \quad \Leftrightarrow \quad \ell_{ij}l_{ik} > c_{i,kj} \quad \Leftrightarrow \quad \ell^2_{jk} > (\ell_{ij} - \ell_{ik})^2$$

so that $\ell_{jk} + \ell_{ik} > \ell_{ij}$. With a permutation of the indices i, j, k we get

$$\begin{cases} \ell_{ij} + \ell_{jk} > \ell_{ij} \\ \ell_{jk} + \ell_{ik} > \ell_{ij} \\ \ell_{ik} + \ell_{ij} > \ell_{ik} \end{cases}$$

c) $m = 3$: For a 3-simplex σ^3 with vertices V_0, V_1, V_2, V_3, the conditions

$$\ell^2_{ij} > 0 \quad , \quad \Omega^2(\sigma^3_{,i}) > 0 \quad , \quad \Omega(\sigma^3) > 0$$

for each i, j, k so that

$$\Omega(\sigma^3) = (\frac{1}{3!}))^2 \begin{vmatrix} \ell^2_{01} & c_{0,12} & c_{0,13} \\ c_{0,12} & \ell^2_{02} & c_{0,23} \\ c_{0,13} & c_{0,23} & \ell^2_{03} \end{vmatrix} > 0 \tag{43}$$

It should be notice that the condition $\Omega(\sigma^3) = 0$, defines the boundary of an open set $\mathcal{D}$ in the space of $\{\ell^2_{ij}\}$ such that if

$$\{\ell^2_{ij}\} \in \mathcal{D} \quad \Rightarrow \quad \{\eta \; \ell^2_{ij}\} \in \mathcal{D} \quad , \quad \forall \eta \in \mathbb{R}_+ \quad . \tag{44}$$

d) $m > 3$: Given the m–simplex σ^m, the edges $\{\ell^2_{ij}\}$ belong to the convex set $\mathcal{D}$ defined by the inequalities (40),(42),(43) and

$$\Omega(\sigma^m) = \frac{1}{(m!)^{-2}} \begin{vmatrix} \ell^2_{01} & \cdots & c_{0,1m} \\ c_{0,12} & \cdots & c_{0,2m} \\ \vdots & \cdots & \vdots \\ c_{0,1m} & \cdots & \ell^2_{0m} \end{vmatrix} > 0 \quad . \tag{45}$$

Also in this case, it is

$$\{\ell^2_{ij}\} \in \mathcal{D} \quad \Rightarrow \quad \{\eta \; \ell^2_{ij}\} \in \mathcal{D} \quad \forall \eta \in \mathbb{R}_+ \quad . \tag{46}$$

so that the edges are homogeneous function.

The volume of the m–simplesso, is an m-degree homogeneous function:

$$\Omega^2(\sigma^m)(\{\eta \; \ell^2_{ij}\}^m_{i,j=0}) = \pm \left(\frac{-1}{2}\right)^m \left(\frac{-1}{m!}\right)^3 \sum_{j_i, k_m} \tilde{\epsilon}^{j_1,\ldots,j_m} \tilde{\epsilon}^{k_1,\ldots,k_m} \prod_{a=1}^{m} \eta \ell^2_{j_a k_a}$$

$$= \eta^m (\Omega\{\ell^2_{ij}\}^m_{i,j=0}) \quad .$$

and, according to the Euler theorem,

$$\sum_{i,j=0}^{m} l^2_{ij} \frac{\partial}{\partial \ell^2_{ij}} \Omega^2(\sigma^m)(\{\ell^2_{ij}\}) = m \; \Omega^2(\sigma^m)(\{\ell^2_{ij}\}) \quad ,$$

so that

$$\sum_{i,j=0}^{m} \ell^2_{ij} \frac{1}{\Omega^2(\sigma^m)} \frac{\partial V^2}{\partial \ell^2_{ij}} = m \quad . \tag{47}$$

or taking into account Eq. (29) $\sum_{ij} \ell^2_{ij} \tilde{g}^{ij} = -m$. by deriving both sides:

$$\frac{1}{\Omega^2(\sigma^m)} \frac{\partial \Omega^2(\sigma^m)}{\partial \ell^2_{kh}} + \sum_{i,j=0}^{m} \ell^2_{ij} \frac{\partial^2 \log \Omega^2(\sigma^m)}{\partial \ell^2_{ij} \partial \ell^2_{kh}} = 0$$

and using Eq. (29), we get

$$\tilde{g}^{kh} = + \sum_{i,j=0}^{m} \ell_{ij}^2 \frac{\partial^2 \log V^2}{\partial \ell_{ij}^2 \partial \ell_{kh}^2} . \tag{48}$$

By deriving (29) and comparing with (48)

$$\tilde{g}^{kh} = + \sum_{i,j=0}^{m} \ell_{ij}^2 \frac{\partial \tilde{g}^{ij}}{\partial \ell_{hk}^2} . \tag{49}$$

that is

$$\frac{\partial \tilde{g}^{hk}}{\partial \ell_{ij}^2} = \tilde{g}^{i(h} \tilde{g}^{k)j} \quad ,$$

being $\tilde{g}^{i(h} \tilde{g}^{k)j} = \frac{1}{2}(\tilde{g}^{ih}\tilde{g}^{jk} + \tilde{g}^{jh}\tilde{g}^{ik})$. There follows that

$$\frac{\partial n_k}{\partial \ell_{ij}^2} = - \left(\frac{1}{2} \right) n_k n^i n^j \tag{50}$$

where $\mathbf{n}^h$ is the normal vector to the face $\sigma^m_{,h}$ opposite to V_h and n_k^h e n^{hk} are its affine components. In fact, according to (36) there results

$$\begin{aligned} \frac{\partial n_k}{\partial \ell_{ij}^2} &= \frac{\partial}{\partial \ell_{ij}^2} [(\tilde{g}^{hh})^{-\frac{1}{2}} \tilde{\delta}_k^h] \\ &= -\frac{1}{2} (\tilde{g}^{hh})^{-\frac{3}{2}} \tilde{g}^{i(h} \tilde{g}^{h)j} \tilde{\delta}_k^h = \\ &= -[\tilde{\delta}_k^h (\tilde{g}^{hh})^{-\frac{1}{2}}][\tilde{g}^{hi} (\tilde{g}^{hh})^{-\frac{1}{2}}][\tilde{g}^{hj} (\tilde{g}^{hh})^{-\frac{1}{2}}] = \\ &= -n_k n^i n^j \end{aligned}$$

7 Self-similar Lattices

In this section some examples of self-similar (scale-invariant) lattices will be given and it will shown that they can be defined by a conformal map of the affine metrics. A self-similar function $f(x)$ is such that $f(\mu x) = \mu^H f(x)$. In the following we will give some self-similar maps defined on 2-simplices.

7.1 Omothety

Let us consider the 2-simplex $\sigma^2 = \{A,\ B,\ C\}$ the map (Fig. 1)

$$\{A,\ B,\ C\} \Longrightarrow \{A,\ B',\ C'\}$$

is such that $\mathbf{n}_C = \pm \mathbf{n}_{C'}$. This is a scale invariant map since there results

$$\ell_{AB}^2 = \lambda \ell_{A'B'}^2 \quad , \quad (0 \leq \lambda).$$

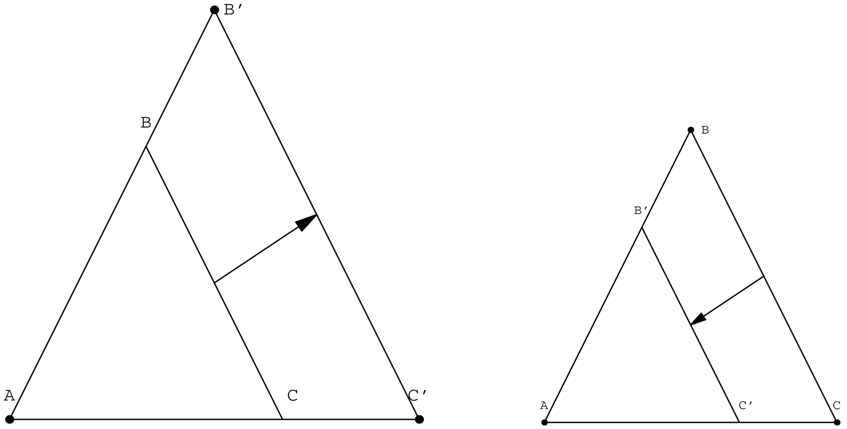


Fig. 1. Omothety map

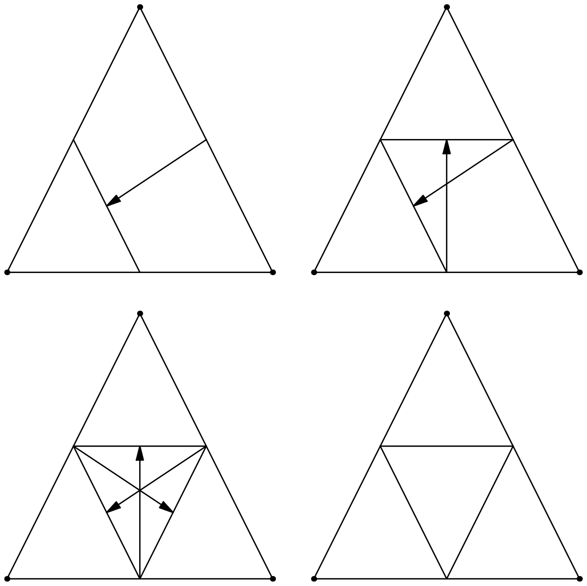


Fig. 2. Fundamental map for the Sierpinski gasket

So that when $\lambda < 1$ we have a contraction, and a dilation when $\lambda > 1$. Due to the invariance of the vector $\mathbf{n}_C$, according to (33), it is also

$$\mathbf{e}^{\prime C} = \lambda \mathbf{e}^C \quad , \quad \mathbf{e}^{\prime A} = \mathbf{e}^A \quad , \quad \mathbf{e}^{\prime B} = \mathbf{e}^B \ .$$

Thus according to (14) the metric $\tilde{g}'_{ij}$ of the transformed simplex is given by a conformal transformation: $\tilde{g}'_{ij} = \lambda \tilde{g}_{ij}$ and from Eq. (25) $\Omega(\sigma^m)' = \pm \lambda^m \Omega(\sigma^m)$.

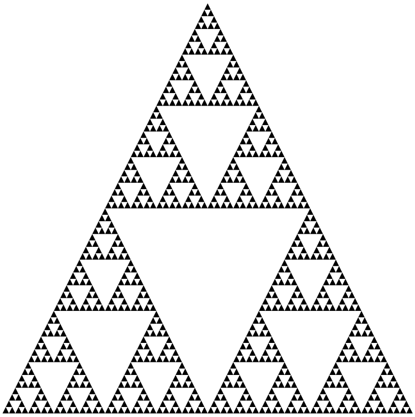


Fig. 3. Sierpinski gasket

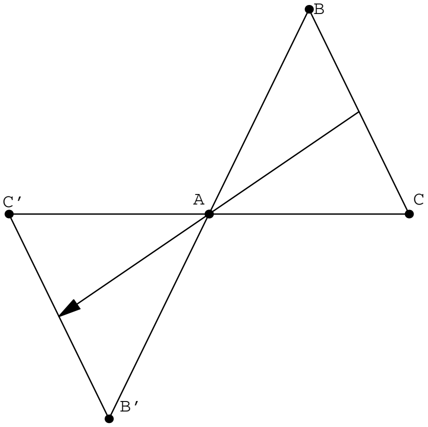


Fig. 4. Husimi Map

7.2 Sierpinski Gasket

The sierpinski gasket, also known as Pascal triangle, can be obtained as a combination of omothety map (Fig. 2) the iterative map will generate the know fractal-shaped curve (Fig. 3).

7.3 Husimi Cacti

As a last example let us consider the map of Fig. 4, which apply a simplex into a counter-oriented equivalent simplex:

$$(\sigma^m)' = -\sigma^m \quad , \quad [A,\ B,\ C] \Rightarrow [A,\ C,\ B] \ .$$

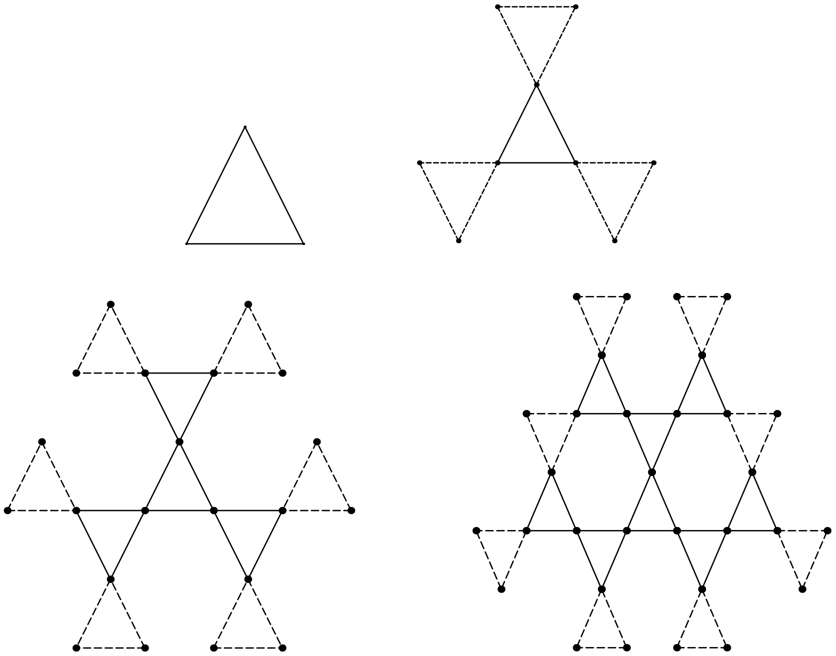


Fig. 5. Husimi cacti

Also in this case the direction of the normal vector is scale-invariant, it changes the orientation. By iterating this map we obtain the Husimi cacti [5] of Fig. 5.

References

1. Bakhoum, E., Toma, C.: Mathematical Transform of Travelling-Wave Equations and Phase Aspects of Quantum Interaction. Mathematical Problems in Engineering 2010 (2010), Article ID 695208, doi:10.1155/2010/695208
2. Barabasi, A.-L.: Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life. Plume Books (2003)
3. Barabasi, A.-L., Ravasz, E., Vicsek, T.: Deterministic scale-free networks. Physica A 299, 559–564 (2001)
4. Boettcher, S., Goncalves, B., Azaret, J.: Geometry and dynamics for hierarchical regular networks. Journal of Physics A 41(33), 335003 (2008)
5. Blumen, A., Bierbaum, V., Mülken, O.: Coeherent dynamics on hierarchical systems. Physica A 371(1), 10–15 (2006)
6. Brewin, L.: A continuous time formulation of the Regge Calculus. Class. Quantum Grav. 5, 839–887 (1988)
7. Cattani, C., Laserra, E.: Symplicial geometry of crystals. J. Interdiscip. Math. 2, 143–151 (1999)
8. Cattani, C., Laserra, E.: Discrete Electromagnetic Action On A Gravitational Simplicial Net. J. Interdiscip. Math. 3, 123–132 (2000)

9. Drummond, I.T.: Regge-Palatini Calculus. Nucl. Phys. 273 B, 125–136 (1986)
10. Fomenko, A.T.: *Differential Geometry and Topology*, Consultant Bureau, New York, London (1987)
11. Gibson, L.J., Ashby, M.F.: Cellular solids: Structure and properties. Pergamon Press, Oxford (1988)
12. Goldenfeld, N.: Lectures on Phase Transitions and the Renormalization Group. Addison-Wesley, Reading (1992)
13. Jourjine, A.N.: Discrete gravity without coordinates. Phys. Rev. D (3) 35(6), 2983–2986 (1987)
14. Kenkre, V.M., Reineker, P.: Exciton Dynamics in Molecular Crystals and Aggregates. Springer, Berlin (1982)
15. Misner, C.W., Wheeler, J.A., Thorne, K.S.: Regge Calculus. In: Gravitation, vol. 42, pp. 1166–1179. W. H. Freeman and Company, New York (1973)
16. Naber, G.L.: Topological Methods in Euclidean spaces. Cambridge University Press, Cambridge (1980)
17. Nishimori, H.: Statistical Physics of Spin Glasses and Information Processing: An Introduction. Oxford Univ. Press, Oxford (2001)
18. Regge, T.: General relativity without coordinates. Il nuovo Cimento 19(3), 558–571 (1961)
19. Singer, I.M., Thorpe, E.J.A.: Lecture Notes on Elementary Topology and Geometry. Scott Foresman and Company, Glenview (1967)
20. Strogatz, S.: Exploring complex networks. Nature 410, 268–273 (2001)
21. Toma, G.: Specific Differential Equations for Generating Pulse Sequences. Mathematical Problems in Engineering 2010 (2010), Article ID 324818, doi:10.1155/2010/324818
22. Vögtle, F. (ed.): Dendrimers. Springer, Berlin (1998)
23. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small' world networks. Nature 393, 440–442 (1998)

Model Driven Software Development of Applications Based on Web Services

Ricardo Rafael Quintero Meza, Leopoldo Zenaido Zepeda Sánchez,
and Liliana Vega Zazueta

Department of Information Systems and Computation
Instituto Tecnológico de Culiacán,
Culiacán, Sinaloa, México
iscrquinter@acm.org, leopoldo@correo.ccs.net.mx,
liliana.vega@salazarvega.net

Abstract. One of the main success factors of the business IT infrastructure is its capacity to face the change. Many companies are defining its IT infrastructure based on Service-Oriented Architecture (SOA), which promises flexibility and efficiency to face the change by reusing and composing loosely coupled services. Because the actual technological platforms used to build SOA systems were not defined originally to this kind of systems, the majority of existing tools for service composition demands that the programmer knows a lot of technical details for its implementation. In this article we propose a conceptual modeling solution to both problems based on the Model-Driven Architecture. Our solution proposes the specification of services and its reuse in terms of platform independent conceptual models. These models are then transformed into platform specific models by a set of Model-to-Model transformation rules, and finally the source code is generated by a set of Model-to-Text transformation rules. Our proposal has been implemented with a tool implemented using the Eclipse Modeling Framework using QVT and Mofscript model transformation languages.

Keywords: Modeling, UML, Service Oriented Architectures, Web-engineering.

1 Introduction

The actual business environment, of high pressure and competition, forces the companies to be efficient and flexible in every aspect of its operation. These efficiency and flexibility is also needed in his IT infrastructure, an important factor that influences its capacity of change and adaptation. For this reason, many companies are defining its technology infrastructure based on the Service-Oriented Architecture (SOA) [1], whose main promise is reuse. This is achieved defining the different business processes as reusable services. The assembly of these services (its choreography) is the way that reuse is achieved.

Different programming models exist to implement reuse in SOA environments, all of them based on the composition of Web services. Although many technological implementation options have been defined (WSFL[2], BPML[3], BPEL[4]), many

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 241–256, 2010.

business continuing using the platforms of traditional programming languages (such Java [5] or .NET [6]). Because in their original definition these platforms were not designed with specific abstractions (instructions or statements) to Web services composition, its definition is accustomed cumbersome (defined using complex libraries or standard APIs), with a lot of technical details that difficult the programmer work and, in consequence, diminishing its capacity to obtain flexibility and adaptability.

This is not a new problem. Since the beginning of electronic computation, the programming languages have been designed having in mind the hiding of low technical details. Although these languages raised the level of abstraction, they still have a "computing-oriented" focus [7]. In other words, they continue expressing concepts in the solution space. In the other hand, raising again the level of abstraction using abstractions defined in terms of the problem space can address the platform complexity. This is the promise of the Model-driven Engineering (MDE) [7], in which the models play first class citizens role.

In MDE there are two basic actors: (1) Domain specific languages (DSL), that define abstractions, relationships and constraints used to construct models for the specification of the space problem and (2) Transformation rules (TR), which define correspondences from the models to software components in the solution space.

From the point of view of the presented problem, we believe that applying MDE in SOA programming environments could be an effective way to improve the work of the designer. In this context, some model-driven solutions exist to generate Web services and its composition in a semi-automatic way [8]; others are similar to the conceptual framework defined by the Web Services Architecture (WSA – [9][10]) and others use UML profiles [11] based on WSDL ([12][13][14]).

With these approaches, the promises of MDE (such full generation of the implementation from the models and model validation) are difficult to achieve because in some cases their models are not established with sufficient details (they have some lack of semantics). In others, the model follows a code visualization approach [15] because it represents specific technology (for example, WSDL) with a level abstraction similar to code. Complexity is another fact to take into account. Because many of the conceptual frameworks designed to specify SOA environments try to capture all the characteristics, they have many model elements not useful in the specification of practical solutions. Some of them only define the first actor of MDE, the DSL, but they do not define the second actor, the TR.

On the other hand, the reuse in SOA is mainly defined in terms of composition. This implies the definition of operations in a new service as an orchestration of operations calls from other services. From the modeling point of view, the specification of the orchestration has been done using dynamic models (mainly expressed as UML Activity diagrams [16]) forgetting the structural aspects. Some technological approaches are based taking into account the structural concerns (such Service Component Architecture – SCA of IBM and Oracle [1]) but in the field of conceptual modeling have not taken this into account in their models.

We consider that structural and dynamical models could be used to define a more complete reuse specification in Web services. With this couple of models, the full code generation task could be also possible. Based on the Model Driven Architecture (MDA) approach [17] this work introduce the following contributions: (1) two models (a Service Model and a Dynamic Model for Service Composition) which allow us to specify the structural and dynamic aspects of reuse in Web service by using aggregation/specialization relationships with a precise semantics, defined using a multidimensional framework [18] (2) a set of TR, defined to obtain the equivalent software artifacts which corresponds with the technological implementation of these models in Java [19] and BPEL [20] platforms and (3) an application of the Web service reuse in the Web Engineering field.

This article is organized as follows: section 2 gives an overview of the MDA approach which defines the model to code strategy in which our proposal is based; section 3 explains details of different platform independent models which capture the structural and dynamic aspects in services; section 4 explains the model to model and the model to code strategy and finally section 5 gives conclusions and further work.

2 The General Development Strategy

The MDA approach is a software development strategy in which the system is described in terms of models. Each model is an abstraction of the system from a different point of view and is used in different stages: to capture the requirements of the system, a Computational Independent Model (CIM) should be defined; to describe the system from a technological independent viewpoint, a Platform Independent Model (PIM) is defined; and finally to describe the system from the technological point of view, a Platform Specific Model (PSM). Each one of these models needs to be specified using a model specification language (or a metamodel). In MDA there are two main approaches for this task: UML profiles[11] or the MetaObject Facility (MOF) [21].

In addition to the models, MDA establishes mapping strategies that enable transformations between them. These mappings transform more abstract models (CIM or PIM) into less abstract models (PSM). The transformation is defined using some model transformation language (QVT[21], ATL[22], UMLX[23], BOTL[24]). This model transformation task is called *Model2Model* [25]. Once the PSM models are obtained, a final code generation strategy needs to be defined to generate the final code. This kind of transformation is called *Model2Text* (or Model2Code [25]) for which several approaches exist (Velocity [26], XDoclet[27], AndroMDA[28], MofScript [29]). The most common is using templates.

Based on the MDA strategy we define PIM and PSM models for service modeling using a MOF approach. A set of *Model2Model* and *Model2Text* transformations were also defined. The *Model2Model* transformations were specified in the QVT language and the *Model2Text* transformations in the MofScript language.

3 Method Overview

The structural and dynamic aspects of the services are specified with two PIM models: a *Service Model (hereafter SM) and a Dynamic Model for Service Composition (hereafter DMSC)*. Fig. 1 shows how these models are related each other. As vertical arrow shows, each one of these models is transformed to a PSM model.

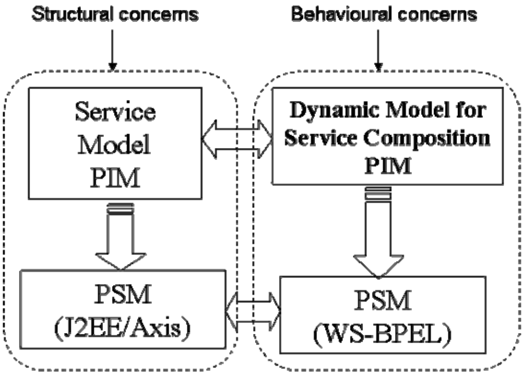


Fig. 1. MDA strategy for service conceptual modeling

Selecting Java, Axis framework and WS-BPEL as examples of technological platforms we define PSM models for them. Because these technologies are large and not all of their constructs needed to implement our PIM models, we define PSM models that include only a subset of the technological platforms needed to generate the software artifacts for the implementation. The horizontal arrows between the PSM models show that both PSM models are interrelated to generate the final code. The knowledge captured in each one is complemented with the knowledge captured in the other model. In this way a full and operative implementation is generated by a set of transformations of the models to code. After that, common compilers are used to obtain the final executable code. The PIM and the PSM metamodels are defined following the MOF approach. In next sections they are introduced and explained.

3.1 Service Metamodel

The Service metamodel defines the PIM. The metamodel has been organized in four related packages: (1) *the Foundation package*, that includes the basic metaclasses of the metamodel. It also includes metaclasses for the data types; (2) *the Structural package*, that includes metaclasses needed to establish aggregation/specialization relationships between services; (3) *the Dynamic package,* that includes metaclasses to specify the behavior of composite services and (4) the *Management package*, which includes metaclasses to organize the services in cohesive groups of services. For space reasons only the Foundation package is explained next.

3.2 Foundation Package

This package includes metaclasses used by other metaclasses in the metamodel. The package is organized in two sub-packages: the *Kernel* and the *Datatype* packages. The *Kernel* package (Fig. 2) includes metaclasses designed to represent the produced and consumed functionality of the application. This functionality is given by Services that include operations and parameters. The produced functionality is represented by a set of *Own-services*. The consumed functionality is represented by a set of *External-services.*

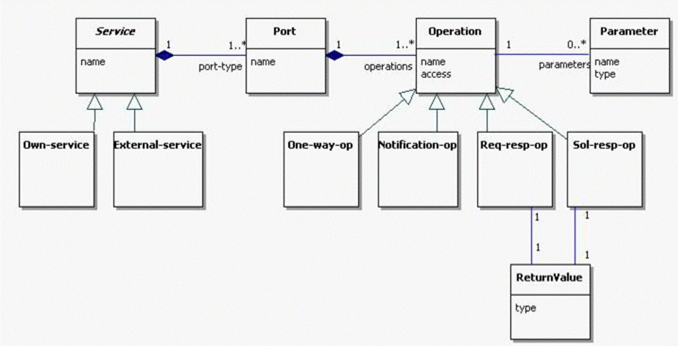


Fig. 2. The Kernel package

We have made this classification because each one has different considerations for service reuse. The basic reason is that in the first case, there is not control over the implementation of the services so the application depends only in the interface definition. In the second case, the application knows both.

Considering the software architecture of the application, the *Own-services* can be defined in two ways: as a view of the business objects operations (from the business layer) and by composition of the public operations from other own or external services. On the other hand, the *External*-services are included in the model by importation based on their interface. For example, if the consumed functionality is implemented by Web services and their interface is defined using WSDL, then a *Text-to-Model* importation process will be needed. Each service has one or more access point called *Ports*. Each port can have one or more operations of the following types:

- **One-way:** an asynchronous operation in the service invoked by a client that does not return a value.
- **Notification:** an asynchronous operation invoked by the service on a client that does not return a value.
- **Request-response:** a synchronous operation in the service invoked by a client that returns a value.
- **Solicit-response:** a synchronous operation invoked by the service on a client that returns a value.

The *DataType* package includes metaclasses for the data types used by other model elements (parameters and return values).

In order to illustrate the application of the Model Service, we are going to use in the conceptual modelling of the Yahoo! Shopping Web application (Fig. 3) [30]. This is an electronic shop where different merchants offer their products. Yahoo! Shopping plays the role of intermediate (like an agent or broker) between the merchants and the clients. The client can search the catalogue of products and select the best. Then the client is redirected to the Web site of the final merchant to buy the selected product. Yahoo! Shopping offers to the developer a Web services API [31]. With this API, the developer can build applications that consume the functionality of Yahoo shopping.

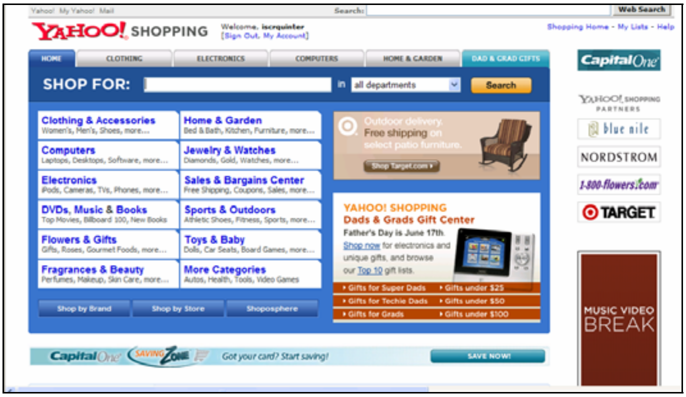


Fig. 3. The Yahoo! Shopping Web application

The set of Web services offered by the application is an example of *Own-services*. The *Service Model* is shown in Figure 4. The model, only an excerpt, shows the following services:

- **ProductSearch:** which allows searching product offerings.
- **MerchantSearch:** which allows retrieving data on a particular merchant or set of merchants.
- **CatalogSpecs:** which allows getting details of a product.
- **UserProductReview:** which allows displaying user reviews about a particular product.

The model also includes the input parameters and the return value of each operation.

3.3 Web Services Structural Relationships

Once the SM includes the different services, the other possibility is to define new services (and consequently new functionality) from them. This new functionality will be given by a new *Own-service* built from the other own or external services. For the complete specification of this new functionality two complementary aspects are considered: *structural* and *dynamic* aspects. In the *structural* aspects, the binding and communication characteristics are captured using aggregation relationships between services. In this case, a new *Own-service* is defined by composition with other services. The relationships are defined in a precise way complementing the model with a set of properties based on a multidimensional framework. In addition to these relationships, *specialization* relationships are also defined. In this case, the relationship is established between a base service and a new derived *Own-service*. The basic idea is that the operations in the base service are specialized in the derived service and also new operations can be added in it.

Following with the Yahoo! Shopping web application, when the client search the catalogue for a specific product then a set of potential merchants with information about the price and a link to its web site is shown (Fig. 5). The client select the best

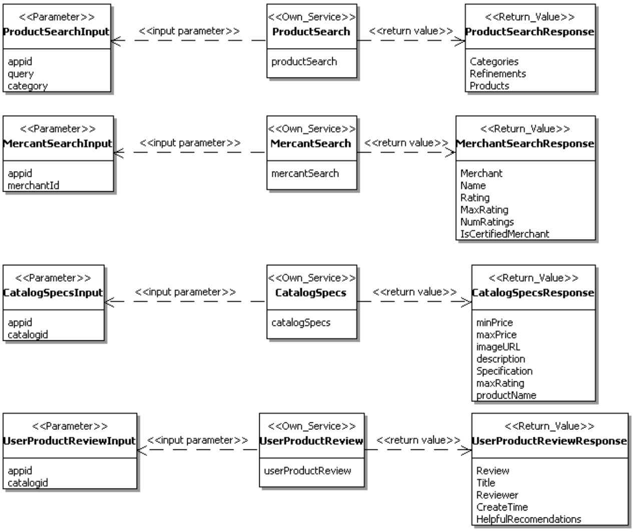


Fig. 4. An excerpt of the Service Model for Yahoo! Shopping

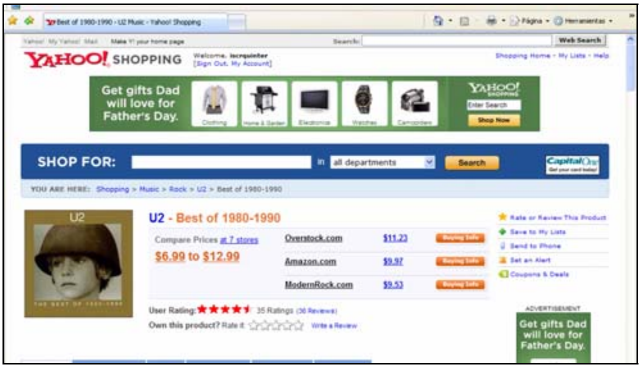


Fig. 5. A product search result in Yahoo! Shopping

price and use the link to transfer the navigation to the web site of the merchant to buy the product.

Some of the merchants are providing Web services APIs to interact with their applications (Amazon[34] , eBay[35]), so we believe that Yahoo! Shopping web application could be improved in these cases if it uses these APIs to allow the client to buy directly the products from it. In this way, Yahoo! Shopping could be act as a façade application to this kind of merchants and also could be an additional business to Yahoo! because many of these merchants offer gains to the clients which use this selling channel.

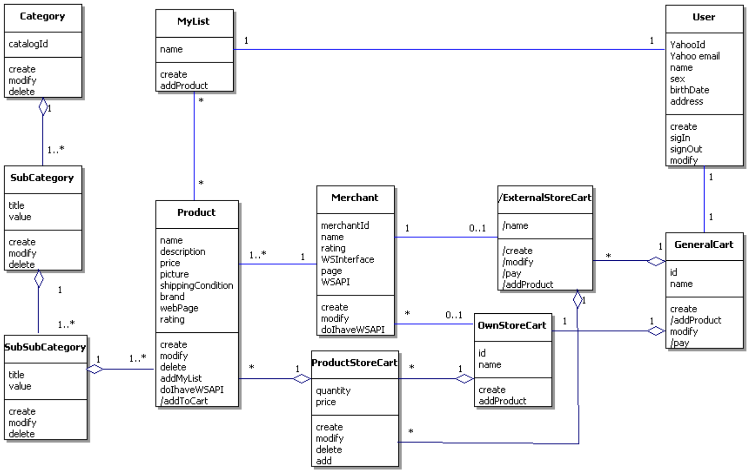


Fig. 6. Business classes in Yahoo! Shopping

In order to show our proposal we show an excerpt of the business classes (with a UML class Diagram) of Yahoo! Shopping (Fig. 6).

We propose to include a *General Cart* business class to represent the shopping cart where the user will put all the articles that he wants to buy from the catalogue. After that, he will go to check out and depending on the kind of store, he could pay directly to the store (in the case it has a Web services API to communicate) or he will be transferred to the Web application (if it has not Web services API) to pay.

The General Cart is an aggregated object that includes two specific kinds of carts. The first one is for representing an external cart (*ExternalStoreCart*). Each one of these carts is for a merchant with a Web services API. The second one is to represent an optional internal cart (*OwnStoreCart*) for merchants who do not have a Web services API. While the user is selecting the different products in Yahoo! Shopping he is putting it in the General Cart. The General Cart has the responsibility of selecting which kind of cart the product will be put depending on if the merchant has or does not have a Web services API.

The functionality of the General Cart could be also exposed as a service and therefore the SM needs to include it (Fig. 7). The *GeneralCartServ* and the *OwnStoreCartServ* are included in the SM as *Own-services*. The *ExternalCartServ* is included as an *External-service* for the cart of a merchant with a Web services API. The *GeneralCartServ* will be a composite service containing as component services the *OwnStoreCartServ* with a static relationship because the internal cart will be unique for all the merchants without a Web services API; and the *ExtStoreCartServ* with a dynamic relationship because it will be selected using dynamic service selection depending on the product.

The *OwnStoreCartServ* is a view of the *OwnStoreCart* business class and the *ExtStoreCartServ* is a set of external services imported into the SM from their interface definition.

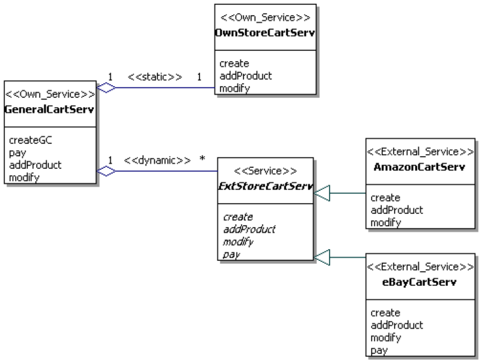


Fig. 7. The General Cart Service in the SM

The *ExtStoreCartServ* is defined as an abstract service (an abstract class), so the specific external services need to be defined as concrete services. Each one of these concrete services will be a service of the mentioned merchants, they are imported to the SM by a *Text2Model* process. This process is defined in such a way that adapts the original interface of the external service to the interface definition in the *ExtStoreCartServ* (with these operations: *create()*, *addProduct()*, *modify()* and *pay()*). For example, figure 7 shows two imported services (concrete external services) for *Amazon* and *eBay*.

In addition to importing the concrete services, the SM requires a selection mechanism that allows the designer to specify which one of the external cart service use, depending on the product that the customer choose. This mechanism is implemented adding a *façade* class into the SM (Figure 8) including an operation used by the designer to select the concrete service. In the model this operation is defined by OCL language and includes a condition (as a formal parameter of the operation) initialized from an expression. As we will show next, this expression comes from used variables in a composite operation definition in the *Dynamic Model for Service Composition* model.

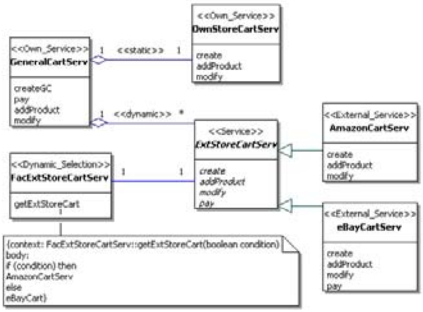


Fig. 8. The dynamic mechanism of dynamic services in the SM

3.4 Service Specialization

Other mechanism available in our proposal is to reuse service functionality by service *specialization*. This is a relationship between an *Own-service* or *External-service* (called the base service) and a new *Own-service* (called the derived service). In this new service some of the operations are specializations of the base service. We based our proposal in the work of Kristensen [36], which defines the attributes and operations of a class as *properties*. In our SM we only have operations as properties, so we define the specialization in their terms.

Following Kristensen[36] four specialization relationship cases are implemented:

- **Intherited property:** the operations in the base service are *inherited* in the new Own-service. With this relationship the operations from the base service are included in the operations of the derived service.
- **Modified property:** the operations in the base service are *refined* in the new derived service. If the base service is an External service, the refinement is different than in the case of an own service, because in the first one is not possible to have control in their implementation so the operation definition can not be modified, however it could be extended.

  This case includes two variants:

  - *Refinement of the signature:* in which the signature of the operation in the base service is refined including one or more additional parameters. The base service can be own or external. Following the *Open-closed principle* [37] the new operation only adds behavior to the original.
  - *Refinement of the operation logic composition:* this case applies to operations defined previously in an *Own-service.* This refinement is carried out by two mechanisms: (i) Identifying in the operation base service *virtual* steps (using the <<virtual>> stereotype). These steps can be refined in the derived service operation; and (ii) Including additional steps in the operation logic of the base service.
- **New property:** in this case a new operation is added in the new derived *Own-service*. The base service can be an Own or External service.
- **Cancelled property:** in which the derived *Own-service* would not include one of the operations in the base service. We do not consider useful this case.

3.5 Service Specialization Metamodel

The metaclasses for the specialization of services are included in the *Kernel* package of the service metamodel (Fig. 9). Because the specialized service is an *Own-service* the derived service (*ODerived-service*) metaclass is a subclass of the *Own-service* metaclass.

There are also metaclasses for the base and derived services. Because the base service could be an Own or External Service, the metamodel includes two metaclasses: the metaclass *OBaseService* (subclass of the *Own-service* metaclass) and the metaclass *EBaseservice* (subclass of the *External-service* metaclass).

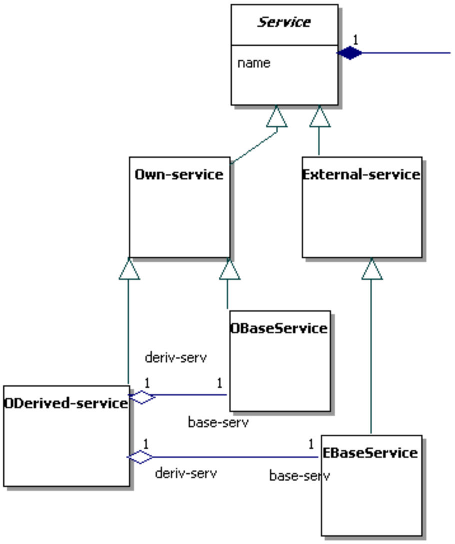


Fig. 9. The service specialization metamodel

Finally, the specialized service (*ODerived-service*) has one and only one of the base service (*OBaseService* or *EBaseService*). This constraint is implemented by a OCL restriction included in the service metamodel.

3.6 Dynamic Package

The logic of each one of the composite service operations is specified with a *Dynamic Model for Service Composition* (DMSC). The DMSC is an UML 2.0 Activity Diagram in which the actions define different control steps and operation invocation of

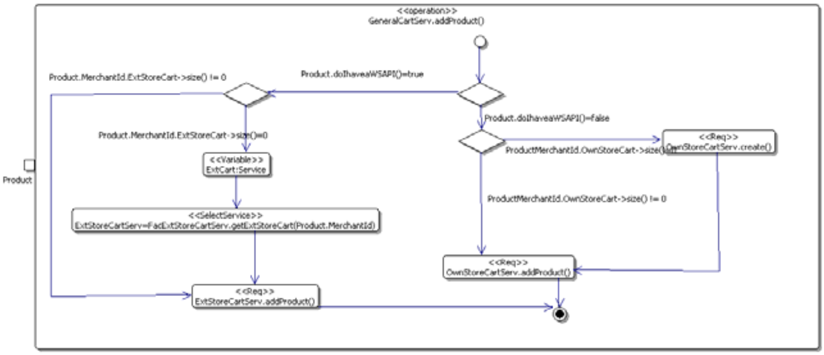

<<operation>>
GeneralCartServ.addProduct()
Product.MerchantId.ExtStoreCart->size() != 0
Product.doIhaveaWSAPI()=true
Product.doIhaveaWSAPI()=false
Product.MerchantId.ExtStoreCart->size()=0
<<Req>>
OwnStoreCartServ.create()
ProductMerchantId.OwnStoreCart->size()=0
<<Variable>>
ExtCart:Service
Product
ProductMerchantId.OwnStoreCart->size() != 0
<<SelectService>>
ExtStoreCartServ=FacExtStoreCartServ.getExtStoreCart(Product.MerchantId)
<<Req>>
OwnStoreCartServ.addProduct()
<<Req>>
ExtStoreCartServ.addProduct()

Fig. 10. The DMSC for `GeneralCart.addProduct` operation

the component services operations. Both, the SM and the DMSC, are complementary views that later are used to generate the implementation code of the new service.

Following with the Yahoo! Shopping!, a DMSC is designed for each one of the own service operations. Focusing only in the the *GeneralCartServ.addProduct()* operation (Figure 10), this is a *One-way* operation that add products selected by the user in the General Cart of Yahoo! Shopping. In the operation definition it is verified if the product store has a Web services API. Depending on this condition, the product is added to an own or external store cart. The operation also verifies if the own or external cart exists, if the cart does not exist then a new car is created. In the case of an external store cart a dynamic service selection is needed.

4 Automatic Code Generation

In order to automatically generate the code from the previous models, we have defined a model transformation strategy based on the Model-driven Architecture (MDA) framework [17]. Starting from the SM and the DMSC models, as *Platform Independent Models* (PIM); and defining *Platform Specific Models* (PSM) for Java, Axis [38] and BPEL technological platforms; a set of model transformations were defined. The following section presents the general strategy.

4.1 General Strategy

The general code generation strategy consists of three basic transformations: (1) *Model2Model*, which transforms the PIM models to PSM models; (2) *Model2Text*, which transforms the PSM models to Java and BPEL source code and finally (3) *Text2Binary*, which generates the final binary code.

The input and output metamodels for the first transformation (*Model2Model*) were defined using the Eclipse Modeling Framework (EMF) [39]. The transformations from the PIM models to the PSM models were defined using the Query-View Transformation (QVT) language [40] (Fig. 11).

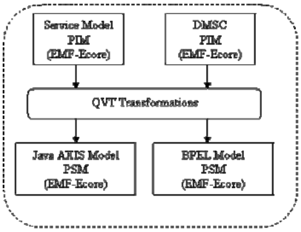


Fig. 11. The *Model2Model* general transformation strategy

The second step of the general strategy defines a set of *Model2Text* transformations, based on templates, from the PSM models to source code. The input of these transformations includes: a Java-Axis PSM metamodel; a BPEL PSM metamodel and a set of templates, which are mixed with the previous metamodels to generate source code for the mentioned platforms. The Java code generated is then compiled with the

Java compiler using the Axis framework and the BPEL code are installed in an application server that executes the final code.

4.2 Model Transformation Scenarios

The general code generation strategy has been organized in four transformation scenarios:

- ***Importing external services*:** which defines the importation from their interface of the external services to the SM.
- ***Generating own services as operation views:*** in which a first transformation generates an Own service from the operations of a business class. A second transformation generates Java classes for the skeleton of the own service and the WSDL interface.
- ***Generation of aggregated and composite services:*** that considers the SM and DMSC as the input of a set of model transformations to the BPEL platform, which implements the composite Web service. Additional code for specific technological details of the selected platform is generated.
- ***Service specialization:*** which includes the code generation strategy of a service specialization. In a first model transformation the base service and the derived service are transformed into a composite service using the Decorator pattern [43]. The composite service generated is then transformed using the third transformation strategy.

Because space reasons we briefly explain the first scenario. This scenario is implemented in three steps. In the first step a SM PIM from the external service is obtained by a *Text2Model* process. In the second step, the SM PIM is transformed in two PSM models by a set of *Model2Model* transformations: one for the Java and the Axis platforms and other for the façade classes. In the third step the final source code is generated from the PSM models obtained in the previous step following the next *Model2Text* algorithm (expressed in natural language):

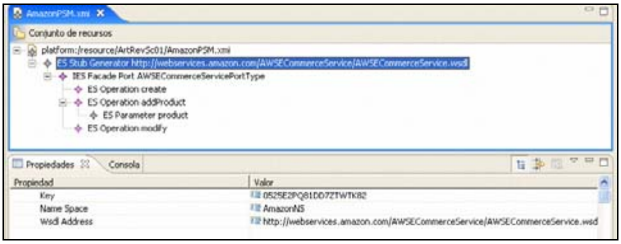
```
For each port of the stub generator:

a. Generate a Java façade:

      1. Generate the support classes (for Java RMI and
      exceptions) and Java
      attributes.
      2. Generate the class constructor with support
      code to access the ports.
      3. Generate the access code to each one of the
      external service operations.
      4. Generate the infrastructure support code.

b. The stub classes are generated from the attribute
values of the stub generator using the wsdl2java tool.
```

As an implementation example of this scenario, the Service-PIM of the *AmazonCartServ* is transformed into the SM-PSM of the Fig. 12.

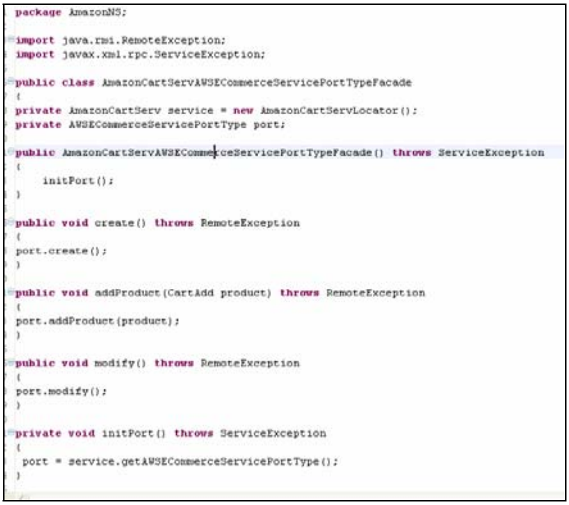


```
platform:/resource/ArtRevSc01/AmazonPSM.xmi
  ES Stub Generator http://webservices.amazon.com/AWSECommerceService/AWSECommerceService.wsdl
    IES Facade Port AWSECommerceServicePortType
      ES Operation create
      ES Operation addProduct
        ES Parameter product
      ES Operation modify
```

Propiedad	Valor
Key	0525E2PQ81DD7ZTWTK82
Name Space	AmazonNS
Wsdl Address	http://webservices.amazon.com/AWSECommerceService/AWSECommerceService.wsd

Fig. 12. The Service-PSM of the *AmazonCartServ* in the EMF editor

Finally, an example an excerpt of the generated code is shown in Fig. 13.



```
package AmazonNS;

import java.rmi.RemoteException;
import javax.xml.rpc.ServiceException;

public class AmazonCartServAWSECommerceServicePortTypeFacade
{
private AmazonCartServ service = new AmazonCartServLocator();
private AWSECommerceServicePortType port;

public AmazonCartServAWSECommerceServicePortTypeFacade() throws ServiceException
{
    initPort();
}

public void create() throws RemoteException
{
port.create();
}

public void addProduct(CartAdd product) throws RemoteException
{
port.addProduct(product);
}

public void modify() throws RemoteException
{
port.modify();
}

private void initPort() throws ServiceException
{
 port = service.getAWSECommerceServicePortType();
}
```

Fig. 13. The generated code for the *AmazonCartServ*

5 Conclusions and Future Work

In this article we have presented a conceptual modeling solution based on the MDA framework to the specification and automatic code generation of services and its reuse. With this approach, the programmer focus his work in the specification of the functionality, as services; and its reuse, using aggregation and specialization relationships between services; instead of focusing in the technical details of the platforms.

We have implemented our solution using the Eclipse EMF framework and the QVT and Mofscript languages. Our tool is implemented using Borland Together Architect 2007 Eclipse version.

As a future work we have to resolve the problem of the dynamic service selection implementation. The actual service oriented platforms do not offer technological facilities for its implementation. For example, the link to component services is fixed in BPEL and is not possible (at least in the original BPEL definition) to change in runtime. We believe that this problem could be solved with the implementation of a technological framework that offers this functionality, but we have not fully implemented this idea.

Other issue we need to resolve is the construction of a graphical modeling editor. Though the EMF automatic editor has been enough for testing our proposal, the construction of bigger models is difficult and error prone with the default editor. This is another aspect that we are going to try in our next research work.

References

1. IBM. The Business Value of the Service Component Architecture (SCA) and Service Data Objects (SDO). White paper (November 2005)
2. Leyman, F.: Web Services Flow Language. Version 1.0. Technical report. IBM (May 2001)
3. Arkin, A.: Business Process modeling language 1.0. Technical report, BPMI Consortium (June 2002), http://www.bpmi.org
4. Andrews, T., et al.: Business Process Execution Language for Web Services. Version 1.1
5. Sun Developer Network (SDN). The source for Java Developers, http://java.sun.com
6. Microsoft. Microsoft.NET, http://www.microsoft.net/net
7. Schmidt, D.C.: Guest Editor's Introduction: Model-driven Engineering. Computer 39(2), 25–31 (2006)
8. Anzbock, R., Dustdar, S.: Semi-automatic generation of Web services and BPEL Processes – A Model-driven approach (Apendix). In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 64–79. Springer, Heidelberg (2005)
9. W3C. Web Services Architecture.W3C Working Group Note 11 February (2004), http://www.w3.org/TR/ws-arch/
10. Massimiliano, C., Elisabetta, D.N., Massimiliano, D.P., Damiano, D., Maurilio, Z.: Speaking a common language: A conceptual model for describing service-oriented systems. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 48–60. Springer, Heidelberg (2005)
11. OMG. UML Profiles and Related Specifications, http://www.uml.org/#UMLProfiles
12. Roy, G., David, S., Ida, S., Jon, O.: Model-driven Web Services Development. In: 2004 IEEE International Conference on e-technology, e-Commerce and e-Service, EEE 2004 (2004)
13. David, S., Roy, G., Ida, S.: Web Service Composition in UML. In: 8th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2004 (2004)
14. Provost, W.: UML for Web Services, http://www.xml.com/lpt/a/ws/2003/08/05/uml.html
15. Brown Alan, W.: Model driven architecture: Principles and practice. Expert's voice. In: Software and Systems Modeling, December 2004, vol. 3(4), pp. 314–327. Springer, Heidelberg (2004) ISSN 1619-1366 (Print) / 1619-1374 (Online)
16. OMG. Unified Modeling Language, http://www.uml.org
17. Object Management Group. OMG Model driven Architecture, http://www.omg.org/mda

18. Albert, M., Pelechano, V., Fons, J., Ruiz, M., Pastor, O.: Implementing UML association, Aggregation and Composition. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, pp. 143–148. Springer, Heidelberg (2003)
19. Sun Developer Network (SDN). Java Platform, `http://java.sun.com`
20. OASIS. WSBPEL, `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel`
21. Object Management Group. MOF 2.0 Query /Views/Transformations RFP, OMG Document: ad/2002-04-10, revised on April 24 (2002)
22. Eclipse. ATL project. ATL Home page, `http://www.eclipse.org/m2m/atl`
23. Willink, E.D.: UMLX: A graphical transformation language for MDA. In: Proceedings of the Workshop on Model Driven Architecture: Foundations and Aplications, University of Twente, Enschede, The Netherlands, June 26-27. 2003. CTIT Technical Report TR-CTIT-03-27. University of Twente (2003), `http://trese.cs.utwente.nl/mdafa2003`
24. Braun, P., Marschall, F.: The Bi-directional Object-Oriented Transformation Language. Technical Report. Technische Universitat Munchen. TUM-I0307 (May 2003)
25. Czarnecki, K., Helsen, S.: Classification of Model Transformation Approaches. OOPSLA 2003 Workshop on Generative Techniques in the Context of Model-Driven Architecture (October 2003)
26. Velocity. The Apache Velocity Project, `http://velocity.apache.org/`
27. XDoclet – Attribute Oriented Programming, `http://xdoclet.sourceforge.net/`
28. AndroMDA, `http://www.andromda.org`
29. Eclipse. MOFScript, `http://www.eclipse.org/gmt/mofscript`
30. Yahoo. Yahoo! shopping, `http://shopping.yahoo.com`
31. Yahoo Developer Network. Yahoo! Shopping Web Services, `http://developer.yahoo.com/shopping`
32. Albert, M., Pelechano, V., Fons, J., Ruiz, M., Pastor, O.: Implementing UML association, Aggregation and Composition. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, pp. 143–148. Springer, Heidelberg (2003)
33. Warmer, J., Kleepe, A.: The object constraint language, 2nd edn. Addison-Wesley, Reading (2003)
34. Amazon.com. Home page: Amazon Web Services, `http://aws.amazon.com`
35. Ebay. eBay Developers Program, `http://developer.ebay.com`
36. Kristensen, B.B., Osterbye, K.: Roles: Conceptual abstraction theory and practical languages issues. Theory and practice of Object Systems 2(3), 143–160 (1996)
37. Meyer, B.: Object-oriented Software construction. IEEE Press, Los Alamitos (1988)
38. Apache Web Services Project. Web Services – Axis, `http://ws.apache.org/axis`
39. Eclipse. Eclipse Modeling Framework (EMF), `http://www.eclipse.org/emf`
40. OMG. MOF QVT Final Adopted Specification, `http://www.omg.org/docs/ptc/05-11-01.pdf`
41. Sun Developer Network. Java Servlet Technology, `http://java.sun.com/products/servlet/`
42. Sun Developer Network. JavaServer Pages Technology, `http://java.sun.com/products/jsp/`
43. Gamma, E., Helm, R., Jonson, R., Vlissides, J.: Design Patterns: elements of reusable object-oriented software. Professional Computing Series. Addison-Wesley, Reading (1995)
44. Oracle Technology Network. Oracle BPEL Process Manager, `http://www.oracle.com/technology/products/ias/bpel/index.html`

An Aspect-Oriented Approach for Mobile Embedded Software Modeling

Yong-Yi FanJiang[1], Jong-Yih Kuo[2], Shang-Pin Ma[3], and Wong-Rong Huang[1]

[1] Department of Computer Science and Information Engineering,
Fu Jen Catholic University, Taipei, Taiwan
{yyfanj,yellow97}@csie.fju.edu.tw

[2] Department of Computer Science and Information Engineering,
National Taipei University of Technology, Taipei, Taiwan
jykuo@ntut.edu.tw

[3] Department of Computer Science and Engineering,
National Taiwan Ocean University, Taipei, Taiwan
albert@mail.ntou.edu.tw

Abstract. Recently, it is one of the most challenging fields in software engineering for embedded software development, since the advancement of embedded technologies has made our life increasingly depend on embedded systems and increased the size and complexity of embedded software. Embedded software developers must pay attention to not only performance and size but also extensibility and modifiability with a view to the complexity rising of embedded software. Besides, one of the characteristics of mobile embedded software is that they are context dependence with crosscutting concerns. Therefore, how to provide a systematic approach to modeling the mobile embedded software, especially on the crosscutting between the sensor, context and reactive behavior, has become an emerging issue in present researches. In this paper, we propose an aspect-oriented modeling process and notations extended from UML for mobile embedded software modeling to deal with the context dependence among sensors and their corresponding reactive functionalities. For the aspect oriented modeling process, the aspects modeling process is provided to separate the concerns of the mobile embedded software. Meanwhile, the extended notations with meta-model framework under class diagram, sequence diagram, and state machine diagram are depicted to facilitate the aspects modeling on structural and behavioral perspectives, respectively. Moreover, a Female Anti-Robbery System is used as an illustrative example to demonstrate our proposed approach.

Keywords: aspect-oriented technique; embedded software; mobile software.

1 Introduction

Traditionally, an embedded system is a special purpose computer (or microprocessor) that is designed and used inside a dedicated device for a particular objective with simplified hardware and primitive software. In recent years, however, because of the

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 257–272, 2010.

advancement of embedded technologies has made our life increasingly depend on embedded systems and hardware grows rapidly, and thus, increased the size and complexity of embedded software [9][27]. There are more and more products such as intelligent mobile phone, car equipment or computer peripherals, which have evolved from single-chip microprocessor to complicated processor containing various peripherals and a small operating system [23]. Besides, embedded software developers must pay attention to not only performance and size but also extensibility and modifiability. In order to decrease the complexity and increase the quality and reusability of embedded software, therefore, how to provide a systematic approach for embedded software development becomes one of the most challenging fields in software engineering [15] [22].

On the hardware, resources of embedded system are limited, and need to be supported by various devices such as camera, RFID and other sensor equipments emphasized in higher performance and reliability [11]. In virtue of the rapidly growing of hardware and complexity of software, however, more and more researchers have to put much emphasis on early stages of development, and to that end, the software development and modeling become more and more important for the development of embedded software [11][17][25].

Dynamic context is one of the characteristics of mobile embedded systems pointed out from [3][11]. The context here means the status of the environment in which the system is operated. Since the location moving on the mobile embedded device is occurrence frequently which causes the status of the environment to change corresponding to the movement, therefore, the behavior of the embedded system is also provided the reactions responding to the environment converted. That is what we say the behavior of the mobile embedded software is dynamic context dependence. For example, the lamp senses the dark and then turns on the light or the handheld device launches the guiding system after the holder enters a museum [2], etc. Kishi and Noda [11] pointed out how to provide a systematic approach to modeling the mobile embedded software, especially on the crosscutting between the context, sensor and reactive behavior, has become an emerging issue.

Numerous approaches [6][7][25][26] had been proposed the aspect-oriented technique for modeling real-time or embedded system, but few of them tackle the crosscutting concerns in the context dependency and lack of providing a systematic process to integrated aspects in the software development processes. In this paper, we attempt to propose an aspect-oriented modeling process and notations extended from UML for mobile embedded software modeling to deal with the context dependence between sensors and their corresponding reactive functionalities. For the aspect oriented modeling process, the aspects modeling processes are provided to separate the concerns of the mobile embedded software. Meanwhile, the extended notations with meta-model framework under class diagram, sequence diagram, and state machine diagram are depicted to facilitate the aspects modeling on structural and behavioral perspectives, respectively

The remainder of the paper is organized as follows: In section 2, we give a brief introduction to aspect-oriented technology. In section 3, we point out our proposed approach on modeling mobile embedded software with aspects and depict the development process for our proposed approach. In section 4, a female anti-robbery system (FARS) served as an illustrated example to demonstrate our approach and the related work is summarized in section 5. We conclude our work in section 6.

2 Approach for Aspect-Orientation

Aspect-oriented software development (AOSD), an emerging development approach, is aimed at promoting improved separation of concerns and increasing comprehensibility, adaptability, and reusability by introducing a new modular unit, called aspects [5][10][27]. It can help us ameliorate modularization by encapsulating crosscutting concerns in separate modular units through the aspect-oriented software development.

2.1 Separation of Concerns

Because of the software functionalities growing greatly, the dependency between the software modules is getting complicated and becomes difficult to understand and maintain. It can help us to identify, encapsulate, and manipulate the complicated software by separating of concerns [12][19][20]. Methods of the separation of concerns could let designers just focus on one concern which they want to analyze or understand the relationships between concerns and become easier to realize and model the whole system. Korpipää [12] pointed out that the main advantages of separated concerns are (1) managing the complexity and enhancing the understandability and traceability throughout the development process to the software; and (2) minimizing the impact of change by proposing encapsulation of different concerns in separate modules.

However, some concerns cannot be separated from the system easily. For example, slogging and security may be occurred in different modules of the system. This concept is called crosscutting concerns. Crosscutting concerns are those that cannot be modularly represented within the selected decomposition [8]. Consequently, their elements are *scattered* (one concern is spread in several modules encapsulating other concerns), and *tangled* (multiple concerns are mixed in one model), together with elements of other concerns [4][20].

2.2 Aspect-Oriented for Crosscutting Concerns

In AOSD, a new unit for modularizing mechanism called *aspects* is used to deal with crosscutting concerns problem [14][21][27]. One of the main elements of an aspect-oriented programming is the *joinpoint* model. The join point describes the “hooks” where enhancements may be added to determine the structure of crosscutting concerns. The patterns that specify the set of join points intercepted for a given application are called as *pointcut*. The implementation of a crosscutting concern in an aspect module is named *advice*. The composition process that injects aspects (or advices) into the join points specified at the point cuts is known as *weaving* process. The aspect system can implement the weaving at compile, load or execution time [1][20][24][25].

Since scattering and tangling problems normally appear throughout the development process, AOSD promotes the separation of crosscutting concerns at every stage of the software lifecycle. Moreover, the explicit representation of crosscutting concerns from the early stages of software development would help us to reason about the modularization of an application design [19].

3 Aspect-Oriented Process and Modeling Notations for Mobile Embedded Software

A number of concerns in mobile embedded software design have the crosscutting relationships impact on the mobile embedded software model. These concerns inherently affect several core modules and their internal modeling elements, like action and context. Examples of crosscutting concerns in mobile embedded software design encompass both internal and systematic properties, such as timing, context, sensor, reaction, liveness, performance and other embedded properties (e.g., power consumption). Without an explicit modeling of such mobile embedded properties, designers cannot properly communicate with and reason about them and their broadly-scoped effects. In this section, we present a meta-modeling framework to support the modular representation of crosscutting concerns in embedded mobile software. The framework is centered on the notion of aspects to describe these concerns including the structural and behavioral aspect modeling extended from the UML which is a de facto standard to the software modeling language.

3.1 Aspects to Embedded Software and Embedded Mobile Software

The principal role of the embedded software is not the transformation of data through the abstract operational functions or procedures, but rather the interaction with the physical world. Lee [13] indicates six characteristics of embedded software, timeliness, concurrency, liveness, reactivity, interface and heterogeneity, distinguishing to the conventional software applications. Besides, Noda *et al.* [17] and Zhang [26] point out that the software size, computing performance, timing constrains and dynamic context, etc. are the most important characteristics on the mobile device. Meanwhile, those characteristics are crosscutting with the core functionality of the embedded software each other. Therefore, inspired the point of view from Lee, Noda *et al.* and Zhang, we propose, in this paper, the aspects to tackle the crosscutting concerns on the embedded software and the more specification on the mobile embedded software.

Aspects of embedded software are the common features that the embedded software is inherited and those features will crosscut with the core functions of the embedded software, and the aspects of embedded mobile software are special concerns that identified from the particular embedded software. Three aspects of

Table 1. Aspects to Embedded Mobile Software

Aspect	Definition
«sensor aspect»	This aspect indicates the various sensing device receiving current data from the environment sensor's values.
«context aspect»	Each context aspect delineates the various statuses of the environment and the transition condition between those contexts.
«reaction aspect»	Reaction aspect defines the possible reactive behaviors corresponding to the specific context.

embedded mobile software, *sensor*, *context* and *reaction* are identified in our approach showed in Table 1. Based on the concepts of the aspects of embedded software and embedded mobile software, in this paper, aspect-oriented software development process, structural meta-modeling framework, and behavioral meta-modeling framework are proposed and delineated in the following subsections, respectively.

In these metamodel frameworks, the white elements are simplifying defined in OMG UML2.2 reference specification [18] and the purple elements are according to aspect concept to be defined in the metamodel by this paper.

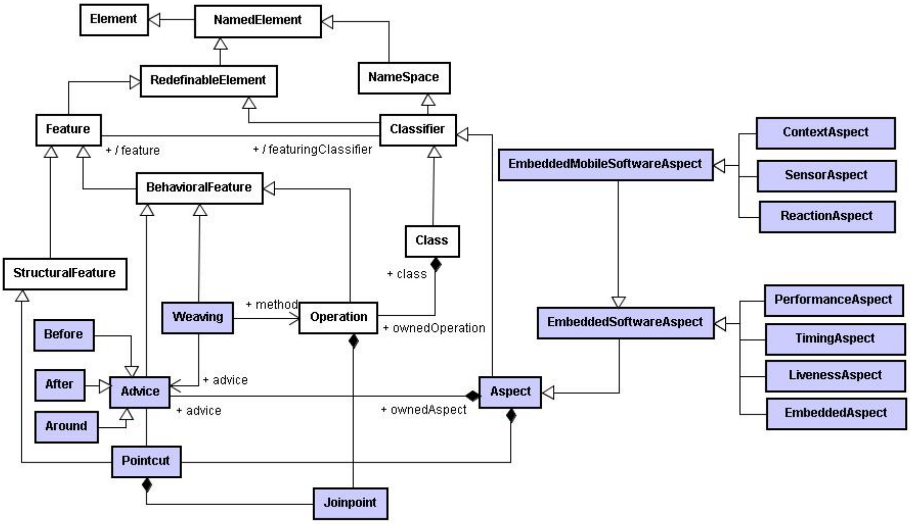


Fig. 1. Structural meta-modeling framework

3.2 Structural Meta-Modeling Framework

Fig. 1 shows the structural meta-modeling framework with aspects for embedded software and embedded mobile software. An element is a constituent of a model and it generates the NamedElement that may have a name. RedefinableElement is redefined in the context of a generation and NameSpace Feature declares a behavioral or structural characteristic of instances of classifiers. *Classifier* describes a set of instances that have features in common. BehaviorFeature and StructureFeature specify an aspect of the behavior of its instances and a typed feature of a classifier, respectively. *Operation* that is contained by Class is inherited from *BehaviorFeature* [18].

Aspect defined in this framework is a kind of *Classifier*, and it is denote the crosscutting concerns in the system. Aspect contains Advice that is a kind of BehaviorFeature and implements the behavior of aspect. Pointcut is a kind of StructureFeature and records the point of weaving. Advice could have three choices to decide the strategy, that are After, Before and Around, respectively. Moreover,

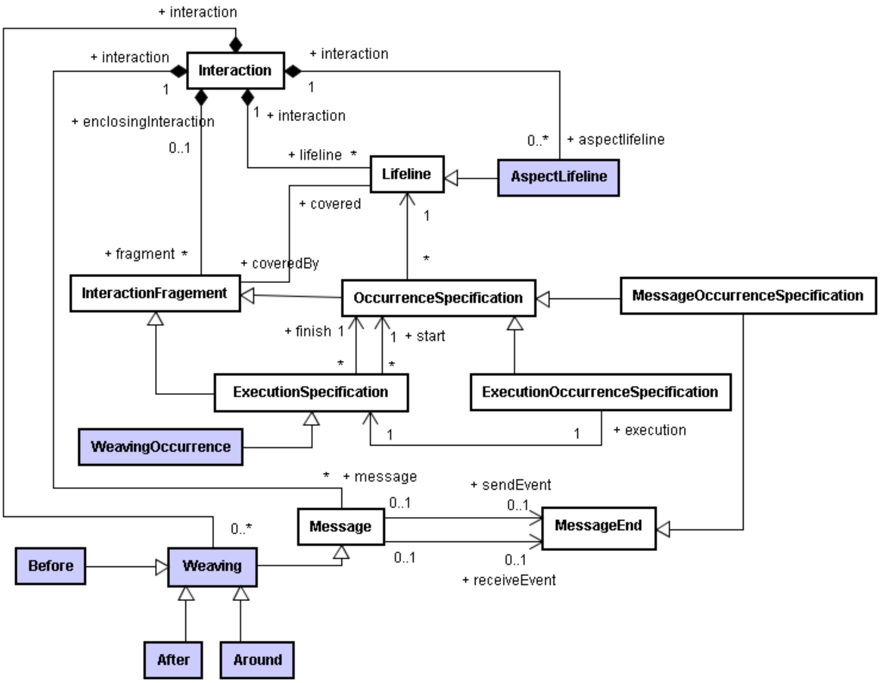


Fig. 2. Behavioral meta-modeling framework

Pointcut is composed by Joinpoint that actually contained by Operation. Weaving element inherits the BehaviorFeature, and it is an abstract metaclass that weaves the Advice into Operation. So, there are the navigable relationship between Weaving, Advice and Operation.

3.3 Behavioral Meta-Modeling Framework

The behavioral meta-modeling framework shown in Fig. 2 is extended from the sequence diagram metamodel to express the weaving interaction between objects and aspects.

In Fig. 2, the Interaction is a behavior element that exchanges information between two connected elements and contains Lifeline that is an individual participant element of interactions. MessageEnd represents what could be occurred at the end of a Message. A message typically associates two EventOccurenceSpecifications, corresponding to sending and receiving events. A Lifeline element denotes the EventOccurenceSpecifications at one of the entities involved in the interaction. An ExecutionSpecification is the specification of the execution of a unit of behavior or action that was execution. ExecutionOccurrenceSpecification represents moments in time at which actions or behaviors start or finish. A MessageOccurrenceSpecification is a kind of MessageEnd [18].

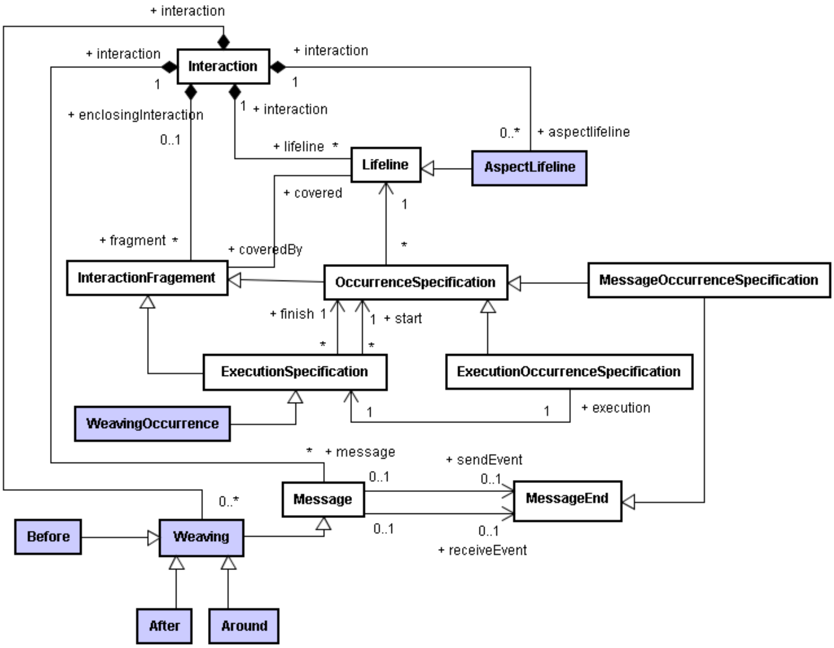


Fig. 3. State Machine meta-modeling framework

AspectLifeline is a kind of Lifeline we extended to represent the aspects event occurrence at one of the aspect involved in the interaction. WeavingOccurrence is kind of ExecutionSpecification and it means the duration of weaving behavior. Weaving is regard as a kind of Message to present the behavior between aspect and core system and it has three kinds of weaving strategy named as Before, After and Around. An aspect may have many advices, but an advice would correspond to one of strategies that consist of after, before and around. Weaving and AspectLifeline are composed in an Interaction.

3.4 State Machine Meta-Modeling Framework

Fig. 3 shows the state machine meta-modeling framework that can be used to express the behavior of part of a system with aspects for mobile embedded software.

The model element StateMachine encompasses different types of transient vertices and Region. State means a situation during which some invariant condition holds. Transition means a directed relationship between a source vertex and a target vertex. FinalState is a special kind of state signifying that the enclosing region is completed, and Vertex that composes Region is an abstraction of a nod. A connection point reference represents a usage of an entry/exit point.

AspectStateMachine including the Advice and Pointcut is composed by Weaving. The Before, After and Around could notify the timing to execute the Advice, i.e. the timing that executes in the (base) StateMachine to transform to Advice in the

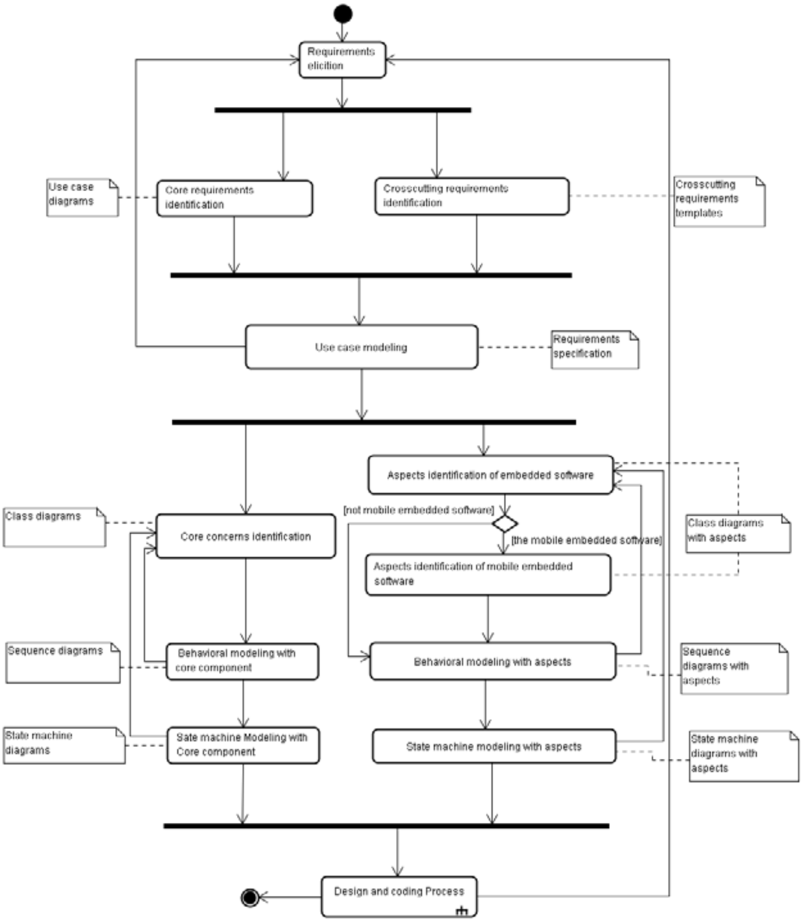

Requirements elicition
Use case diagrams
Core requirements identification
Crosscutting requirements identification
Crosscutting requirements templates
Use case modeling
Requirements specification
Aspects identification of embedded software
[not mobile embedded software]
[the mobile embedded software]
Class diagrams
Core concerns identification
Class diagrams with aspects
Aspects identification of mobile embedded software
Sequence diagrams
Behavioral modeling with core component
Behavioral modeling with aspects
Sequence diagrams with aspects
State machine diagrams
Sate machine Modeling with Core component
State machine modeling with aspects
State machine diagrams with aspects
Design and coding Process

Fig. 4. Aspect-oriented Software development process to embedded software

AspectStateMachine. The WeavingState is a kind of State to show the state in the (base) StateMachine that means execute flow has transformed to AspectStateMachine.

3.5 Aspect-Oriented Software Development Process

According to the structural and behavioral meta-modeling framework described above, an aspects-oriented software development process for embedded mobile software is depicted in Fig. 4. This process can be divided into two parts. The first one is the requirements engineering including the functional requirements elicitation and crosscutting requirements identification. The other one is system modeling which comprises core system modeling and embedded software modeling with aspects.

Requirements Elicitation– in this activity, the software requirements are elicited through problem statements or documents provided by the customers, developers or stakeholders. The requirements elicitation activity is the starting point of the software development and can be realized by the core functions and other aspects.

Core requirements and crosscutting requirements identification – in this phase, the core requirements are modeled by using the use case diagram to describe the system's functional requirements as well as the crosscutting requirements are identified to provide the basis for aspects modeling.

Use case modeling – after the functional and crosscutting requirements are identified, the use case model served as the core component of software requirements specification are specified. Detailed scenarios of each use case are designated and crosscutting relationship between use cases are pointed out.

Core concerns modeling – in this activity, the core functional concerns are identified served as the basis for core modeling. The class diagram is used to model the main concepts and corresponding relationships of the system.

Behavioral and state machine modeling to core concerns – in these tow activities, sequence diagrams and state machine diagrams are used to delineate the system's dynamic interaction and state transition. These models could describe the system's behavioral perspectives.

Aspects identification of embedded software – in this activity, aspects are identified according to crosscutting requirements described in CRT. Each crosscutting concerns is the candidate aspects in the system. In this activity, designer can model the class diagram with aspects according the structural metamodel framework which is extended class diagram with aspects. If the target system is belonging to embedded mobile system, the aspects of embedded mobile software are modeled followed by this activity; otherwise, if the target system is not belonging to mobile software, the aspects identification to mobile software activity could be skipped over.

Aspects identification of embedded mobile software – three kinds of structural mobile aspects, context aspect, sensor aspects, and reaction aspect, are identified and modeled in this activity by using the extended class diagram with aspects described in section 3.2.

Behavioral and state machine modeling with aspects – in this activity, extended sequence and state machine diagrams mentioned in section 3.3 and 3.4, respective, are used to depict the weaving process between core classes and aspects to model the system's dynamic behavior.

4 Female Anti-Robbery System (FARS)

The FARS we used to demonstrate our proposed approach is mobile embedded software that can provide the anti-robbery and anti-thief functionalities operated on a smart phone with g-sensors and GPS device.

According to the aspect-oriented embedded software development process proposed in section 3.5, the use cases model is specified firstly to represent the

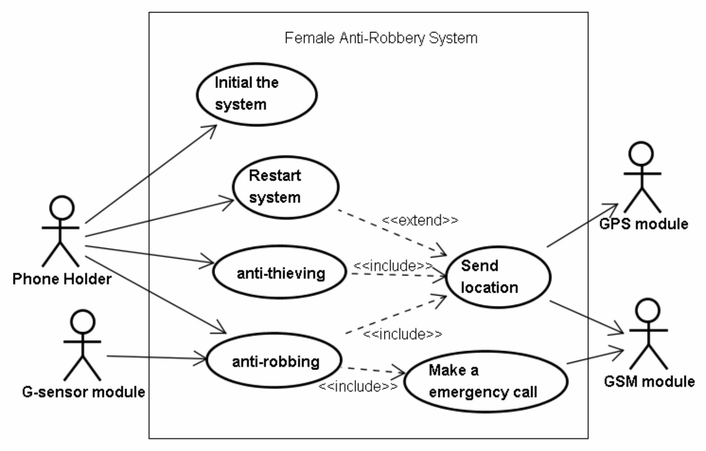


Fig. 5. Use cases diagram for FARS

system's core functional and crosscutting requirements. The use case diagram of FARS system is shown in Fig. 5. In this system, four actors are identified named Phone Holder, G-sensor device, GPS device, and GSM device. The Phone Holder can set up the system, restart system and perform the anti-thieving and anti-robbing functionalities. Since the sending location provided by the GPS and GSM device are the common part of anti-thieving and anti-robbing use cases, the Send location is isolated as an use case through the include stereotype.

4.1 Aspects of Embedded Mobile Software on FARS

The whole model of the system of FARS is separated into two parts: core model and aspects model. The core model presents the basic functionalities of the system, and the aspects model specifies the embedded mobile software aspects. Aspects of embedded software modeling are the general aspect for any kinds of mobile embedded software, and the aspects of embedded mobile software are special concerns identified from the particular embedded software.

In this example, we will concentrate only on the context dependency modeling of embedded mobile software aspects. The core model contains three classes named SituationHandling, SensorInfo, and ActionsHandling, which are crosscut among contexts, sensors, and reactions using the «weaving» stereotype, respectively. We use composite structure diagram of UML 2.2 for this modeling.

4.2 Structural Aspects Modeling on FARS

The structural aspect modeling is based on the aspects of embedded mobile software analysis to model the corresponding crosscutting concerns, pointcut, advice, and weaving relationships.

In this case, three aspects of context aspect named Normal Context, CaseJudgment Context, and Abnormal Context are identified, and the aspects on context, sensor, and reaction aspects are modeled in Fig. 6.

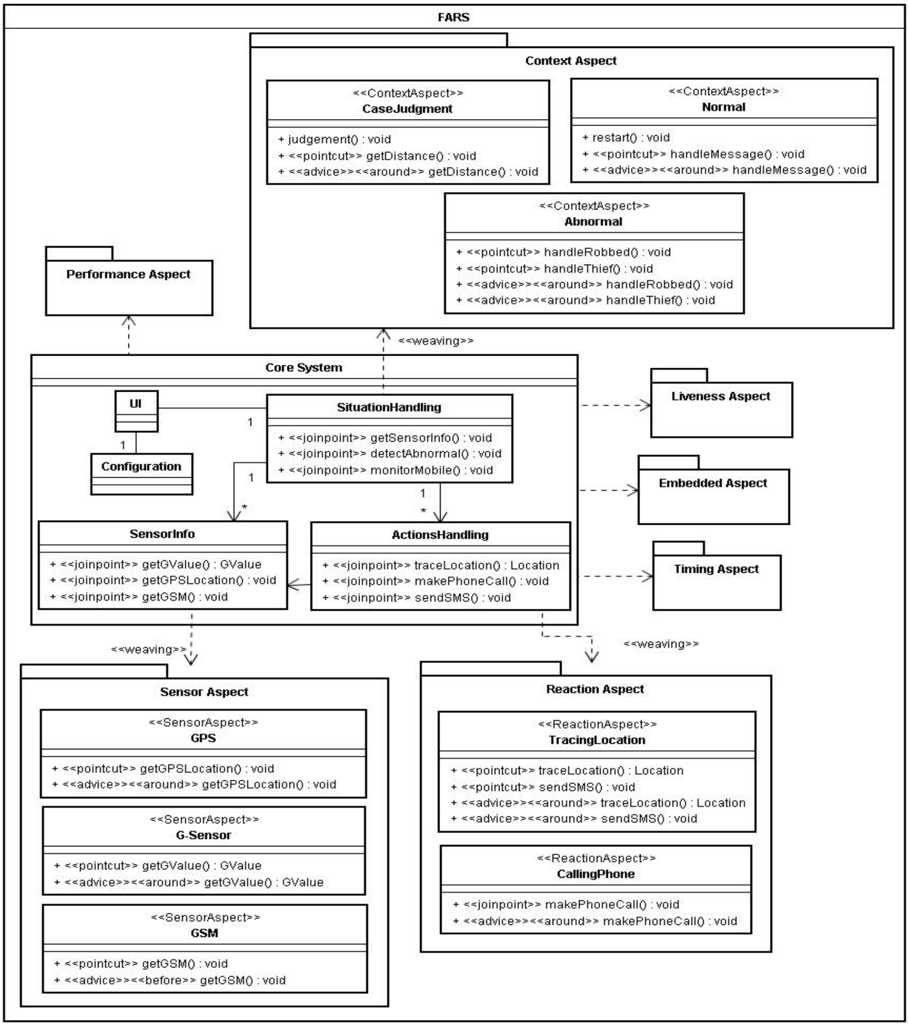

FARS
Context Aspect
<<ContextAspect>>
CaseJudgment
+ judgement() : void
+ <<pointcut>> getDistance() : void
+ <<advice>><<around>> getDistance() : void
<<ContextAspect>>
Normal
+ restart() : void
+ <<pointcut>> handleMessage() : void
+ <<advice>><<around>> handleMessage() : void
<<ContextAspect>>
Abnormal
+ <<pointcut>> handleRobbed() : void
+ <<pointcut>> handleThief() : void
+ <<advice>><<around>> handleRobbed() : void
+ <<advice>><<around>> handleThief() : void
Performance Aspect
<<weaving>>
Core System
UI
1
Configuration
SituationHandling
+ <<joinpoint>> getSensorInfo() : void
+ <<joinpoint>> detectAbnormal() : void
+ <<joinpoint>> monitorMobile() : void
1
1
1
*
*
SensorInfo
+ <<joinpoint>> getGValue() : GValue
+ <<joinpoint>> getGPSLocation() : void
+ <<joinpoint>> getGSM() : void
ActionsHandling
+ <<joinpoint>> traceLocation() : Location
+ <<joinpoint>> makePhoneCall() : void
+ <<joinpoint>> sendSMS() : void
Liveness Aspect
Embedded Aspect
Timing Aspect
<<weaving>>
<<weaving>>
Sensor Aspect
<<SensorAspect>>
GPS
+ <<pointcut>> getGPSLocation() : void
+ <<advice>><<around>> getGPSLocation() : void
<<SensorAspect>>
G-Sensor
+ <<pointcut>> getGValue() : GValue
+ <<advice>><<around>> getGValue() : GValue
<<SensorAspect>>
GSM
+ <<pointcut>> getGSM() : void
+ <<advice>><<before>> getGSM() : void
Reaction Aspect
<<ReactionAspect>>
TracingLocation
+ <<pointcut>> traceLocation() : Location
+ <<pointcut>> sendSMS() : void
+ <<advice>><<around>> traceLocation() : Location
+ <<advice>><<around>> sendSMS() : void
<<ReactionAspect>>
CallingPhone
+ <<joinpoint>> makePhoneCall() : void
+ <<advice>><<around>> makePhoneCall() : void

Fig. 6. Structural Aspects modeling for FARS

Context indicates any environment in which the system is operated and includes much information. We focus on anti-robbing subsystem to set three contexts for our system. According to the mobile status, Normal context and Abnormal context is depicted. And most of time, G-motion sensor need to detect the value and *CaseJudgment* context is adding in.

In Fig. 6, we extend the operations of core functions with «join point» stereotype to present the join point relation in AOSD as well as «advice» stereotype to present the implement operations of advice. The weaving process is extended from the dependency relations with «join point» stereotype.

For example, in this case, the SensorInfo class has a join point within the operation *getGValuew*. When the *getGValue* operation is executed and triggered by the join point, the weaving process is activated and the corresponding advice *getGValue* implemented in the G-sensor aspect is composed (static or dynamic) into the SensorInfo class. The composite structure diagram is used to organize the structure of those aspects; furthermore, the dynamic behavior is exhibited in section 4.3.

4.3 Behavioral Aspects Modeling on FARS

We use two models, sequence diagram and state machine diagram, to specify the behavioral perspective for a mobile embedded software modeling with aspects. The sequence diagram is used to depict the interaction relationships among the core function objects as well as the weaving relationships between the core function object and aspect. The state machine diagram is used to present the internal states and transitions of an aspect object.

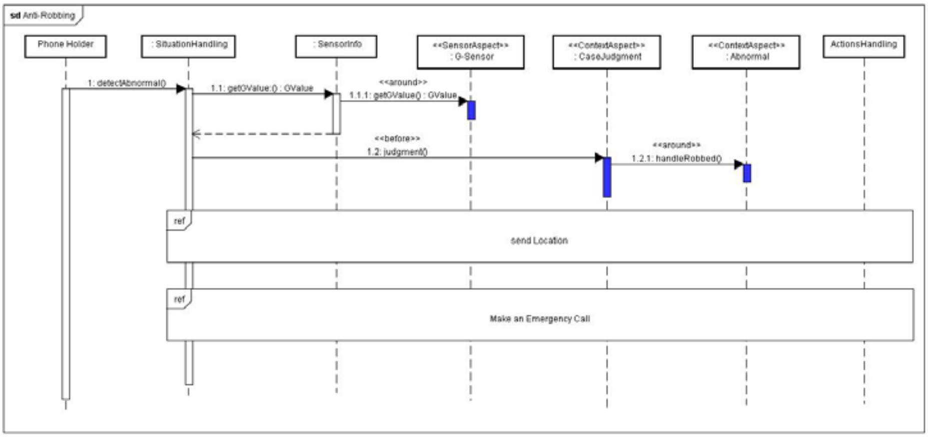


Fig. 7. Sequence diagram for anti-robbing use case

Sequence diagram. Fig. 7 shows the sequence diagram for the anti-robbing and Send location use case specified in Fig. 5, respectively. We provide two extensions to a sequence diagram. The first one is the «weaving» stereotype on the interaction for presenting the composition relationships weaving the aspect into the core function through a join point. The other one is the advice activation extended from the activation bar (blue activation in Fig. 7) to indicate the advice implementation and execution.

State machine diagram. Fig. 8 shows the states and state transitions of the context aspect using the state machine diagram. In this diagram, there are three nested state diagrams to present the aspects. Besides, we extend the **do** action in a state with «advice» stereotype to present the weaving signal and action. In, Fig. 8 "*check angle changed*" state in the *CaseJudgment* aspect is to calculate the angle variability through the special sensor. This sensor can be g-sensor (used in this case), camera or others; hence, for decreasing the coupling among the context, sensor, and reaction, an

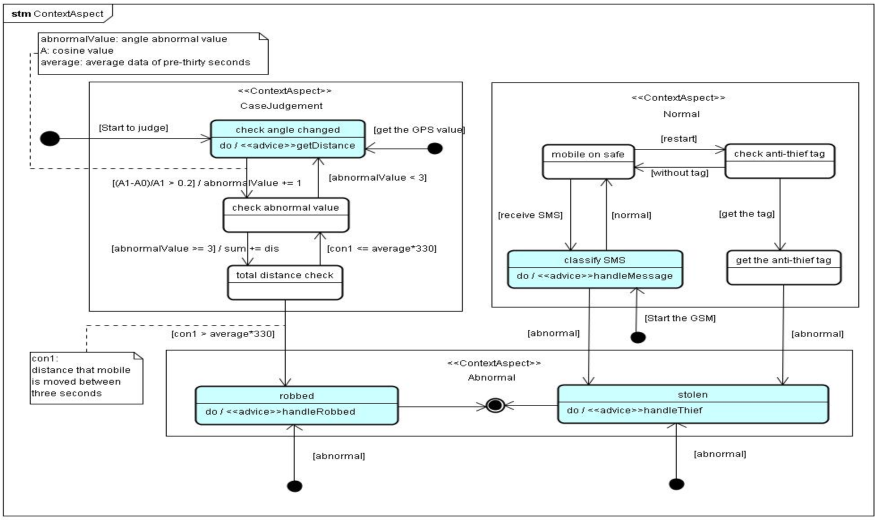


Fig. 8. State machine diagram for context aspects

aspect is separated, and the weaving signal and action is attached to the **do** action to present the weaving behavior with the core model of a state. Other examples are shown in *robbed* and *stolen* state, respectively.

5 Related Work

M. Mousavi *et al.* [16] provide a design framework based on the multi-set transformation language called GAMMA. The proposed method consists of separating the aspects of functionality, coordination, timing and distribution in the design phase, and providing a weaving mechanism to provide a formal semantics for composed aspects. The weaving method enables the designer to have localized reasoning about the properties of aspect models and their inter-relationships. GAMMA language may help developer to analysis the embedded real-time system. But it has the drawback if developers did not learn GAMMA before. They need spent more time to understand GAMMA. Otherwise model graphs are friendlier to stakeholders than GAMMA language.

Natsuko Noda and Tomoji Kishi [11][17] focus on one of the important characteristics of embedded system called context dependency, and examine problems in context modeling. They define three types of aspects, process, context and sensor, each of them corresponds to the conceptual part of context dependent system. A new modeling element called inter aspects relation are explicitly defined to specify the relationship among aspects. Although we also focus on context property of embedded system we more emphasize on the mobile embedded system that gives full play to this characteristic. Using the OMG UML spec. is more general then them, too. It is easy to follow our modeling approach to model the new system.

Lichen Zhang [25] presents an aspect-oriented method to model the embedded real-time system based on UML and describes the real-time features as an independent aspect in order to make the multimedia system easier to design and develop and guarantee the time constraints. In the other hand, we can say it is a narrow sense to other system because it only emphasize on time aspect. Moreover it is vague of its modeling notations. Although there are provided some extension semantics in the paper. But that syntax would not clearly than diagram notations.

Freitas *et al.* [7] present a proposal to use aspect orientation in the analysis and design of Distributed Real-time Embedded (DRE) systems. They performed an adaptation of a well-defined method called FRIDA (From Requirements to Design using Aspects), which was originally applied to the fault tolerant domain. The proposed adaptation includes the use of RT-UML together with aspect-oriented concepts in design phase, aiming to separate the handling of non-functional from functional requirements. In this paper, it only focuses on the non-functional requirement but the problem of crosscutting concern not only happened on the non-functional requirements but also the functional requirements probably.

6 Conclusion

In this paper, we proposed an aspect-oriented modeling process and notations extended from UML for mobile embedded software modeling to deal with the context dependence between sensors and their corresponding reactive functionalities. For the aspect oriented modeling process, the embedded and embedded mobile aspects modeling processes were provided to separate the concerns of the mobile embedded software. Meanwhile, the extended notations with meta-model framework under class diagram, sequence diagram, and state machine diagram were depicted to facilitate the aspects modeling on structural and behavioral perspectives, respectively. Moreover, a Female Anti-Robbery System was used as an illustrative example to demonstrate our proposed approach.

Acknowledgments

The research is partially sponsored by National Science Council (Taiwan) under the grant NSC97-2221-E-030-009 and NSC98-2220-E-030-003, and by Fu Jen Catholic University (Taiwan) under the grant FJU409831044041.

References

1. Agostinho, S., Ferreira, R., Moreira, A., Marques, A., Brito, I., Kovačević, J., Araújo, J., Raminhos, R., Ribeiro, R.: A Metadata-driven Approach for Aspect-oriented Requirements Analysis. In: the 10th International Conference on Enterprise Information Systems (ICEIS 2008), Barcelona, Spain, June 12-17, (2008)
2. Aldawud, T., Bader, A.: UML Profile for Aspect-oriented Software Development. In: The Third International Workshop on Aspect Oriented Modeling (2003)

3. Altisen, K., Maraninchi, F., Stauch, D.: Aspect-oriented Programming for Reactive Systems: Larissa, A proposal in the Synchronous Framework. Science of Computer Programming 63(3), 297–320 (2006)
4. van den Berg, K.G., Conejero, J.M., Chitchyan, R.: AOSD Ontology 1.0 Public Ontology of Aspect-oriented. Technical Report AOSD-Europe Deliverable D9, AOSD-Europe-UT-01 (2005)
5. Filman, R., Elard, T., Clarke, S., Aksit, M.: Aspect-oriented Software Development. Addison-Wesley (2004)
6. Freitas, E.P., Wehrmeister, M.A., Pereira, C.E., Wagner, F.R., Silva Jr., E.T., Carvalho, F.C.: DERAF: a High-level Aspects Framework for Distributed Embedded Real-time Systems Design. In: 10th International Workshop on Early Aspects, pp. 55–74. Springer, Heidelberg (2007)
7. Freitas, E.P., Wehrmeister, M.A., Pereira, C.E., Wagner, F.R., Silva Jr, E.T., Carvalho, F.C.: Using Aspects to Model Distributed Real-time Embedded Systems. In: Workshop on Aspect-oriented Software Development, Florianopolis, Brazil (2006)
8. Fuentes, L., Gámez, N., Sánchez, P.: Aspect-oriented Executable UML Models for Context-aware Pervasive Applications. In: The 5th International Workshop on Model-based Methodologies for Pervasive and Embedded Software (MOMPES 2008), pp. 34–43 (2008)
9. Graaf, B., Lormans, M., Toetenel, H.: Embedded Software Engineering: the State of the Practice. IEEE Software 20(6), 61–69 (2003)
10. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C.V., Loingtier, J.-M., Irwin, J.: Aspect-oriented Programming. In: Aksit, M., Matsuoka, S. (eds.) ECOOP 1997. LNCS, vol. 1241, pp. 220–242. Springer, Heidelberg (1997)
11. Kishi, T., Noda, N.: Aspect-oriented Context Modeling for Embedded System. The the Workshop on Aspect-Oriented Requirements Engineering and Architecture Design, Early Aspects (2004)
12. Korpipää, P.: Blackboard-based Software Framework and Tool for Mobile Device Context Awareness. PhD thesis. VTT Publications 579 (2005)
13. Lee, E.A.: Embedded Software. In: Zelkowitz, M. (ed.) Advances in Computers, vol. 56, Academic Press, London (2002)
14. Lee, J.-S., Bae, D.-H.: An Aspect-oriented Framework for Developing Component-based Software with the Collaboration-based Architectural Style. Information and Software Technology 46(2), 81–97 (2004)
15. Liggesmeyer, P., Trapp, M.: Trends in Embedded Software Engineering. IEEE Software 26(13), 19–25 (2009)
16. Mousavi, M., Russello, G., Cursaro, A., Shukla, S., Gupta, R., Schmidt, D.C.: Using Aspect-GAMMA in the Design of Embedded Systems. In: The 7th Annual IEEE International Workshop on High Level Design Validation and Test. IEEE Computer Society Press, Los Alamitos (2002)
17. Noda, N., Kishi, T.: Aspect-oriented Modeling for Embedded Software Design. Japan Advanced Institute of Science and Technology (2007)
18. Object Management Group, UML 2.2 Superstructure Specification, `http://www.uml.org`
19. Ortin, F., Cueva, J.M.: Dynamic Adaptation of Application Aspects. Journal of Systems and Software 71(3), 229–243 (2004)
20. Ortin, F., Cueva, J.M.: Malaca: A Component and Aspect-oriented Agent Architecture. Information and Software Technology 51(6), 1052–1065 (2009)

21. Ray, I., France, R., Li, N., Georg, G.: An Aspect-based Approach to Modeling Access Control Concerns. Information and Software Technology 46(9), 575–587 (2004)
22. Voelter, M., Salzmann, C., Kircher, M.: Model Driven Software Development in the Context of Embedded Component Infrastructures. In: Atkinson, C., et al. (eds.) Component-Based Software Development for Embedded Systems. LNCS, vol. 3778, pp. 143–163. Springer, Heidelberg (2005)
23. Wehrmeister, M.A., Freitas, E.P., Pereira, C.E., Ramming, F.: GenERTiCA: A Tool for Code Generation and Aspects Weaving. In: 11th IEEE International Symposium on Object, component, serrvice-oriented Realtime Distributed Computing (ISORC 2008). Springer, Heidelberg (2008)
24. Yamada, K., Watanabe, T.: An Aspect-oriented Approach to Modular Behavioral Specification. Electronic Notes in Theoretical Computer Science 163(1), 45–56 (2006)
25. Zhang, L.: Aspect-oriented Analysis for Embedded Real-time Systems. In: Advanced Software Engineering & Its Applications (ASEA), pp. 53–56. IEEE Press, Los Alamitos (2008)
26. Zhang, Q., Zhang, L.: Aspect Oriented Middleware for Mobile Real-time Systems. In: Advanced Software Engineering & Its Applications (ASEA), pp. 138–141 (2008)
27. Zhu, Z.J., Zulkernine, M.: A Model-based Aspect-oriented Framework for Building Intrusion-aware Software Systems. Information and Software Technology 51(5), 865–875 (2009)

Multi-variate Principal Component Analysis of Software Maintenance Effort Drivers

Ruchi Shukla and A.K. Misra

Computer Science and Engineering Department
Motilal Nehru National Institute of Technology, Allahabad, India – 211004
ruchishukla_mtech@rediffmail.com, akm@mnnit.ac.in

Abstract. The global IT industry has already attained maturity and the number of software systems entering into the maintenance stage is steadily increasing. Further, the industry is also facing a definite shift from traditional environment of legacy softwares to newer softwares. Software maintenance (SM) effort estimation has become one of the most challenging tasks owing to the wide variety of projects and dynamics of the SM environment. Thus the real challenge lies in understanding the role of a large number of SM effort drivers. This work presents a multi-variate analysis of the effect of various drivers on maintenance effort using the Principal Component Analysis (PCA) approach. PCA allows reduction of data into a smaller number of components and its alternate interpretation by analysing the data covariance. The analysis is based on an available real life dataset of 14 drivers influencing the effort of 36 SM projects, as estimated by 6 experts.

Keywords: Software maintenance, Effort estimation, Principal Component Analysis.

1 Introduction

Software maintenance (SM) is defined as ‘the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a changed environment’ [1]. SM is just not the last phase of software development life cycle but it an iterative process. SM is a structured but complicated, expensive and time consuming process, particularly in case of legacy and large systems. SM is a dynamic process and problems of maintainer’s turnover, recruitment of experienced personnel, maintenance bid costing and optimum resource allocation have made accurate estimation of SM cost a fairly challenging problem [2].

2 Literature Review

The popular datasets of software development and maintenance include - COCOMO 81, COCOMO II, Rao and Sharda, COSMIC, IFPUG, Desharnais, Kemerer etc. ([3], [4]). International Software Benchmarking Standards Group (ISBSG-2005) provides an analysis of the maintenance and support dataset. Recent research has focused on

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 273–284, 2010.

the use of function points (FPs) in effort estimation. However, a precise estimation should not only consider the FPs representing the software size, but should also include different elements of the development environment. Reference [5] proposed a SM project effort estimation model based on FPs. Reference [6] and [7] listed the following four groups of factors affecting the outsourced maintenance effort: system baseline, customer attitude, maintenance team and organizational climate; and described how a system dynamics model could be build.

The unit effort expended on maintenance of a system is dependent on many external factors and is not a linear relation with respect to time [8]. Hence, various artificial intelligence (AI) based techniques like artificial neural networks (ANN), genetic algorithms (GA), fuzzy logic (FL), case based reasoning etc, and regression analysis have been used for accurate estimation of SM effort ([9]-[10]). Most of these studies are based on the FP metrics or the 'maintained lines of code' metric, which is difficult to estimate. Reference [11] compared the prediction accuracy of different models using regression, neural networks and pattern recognition approaches**.** To model the conditions typically present in a modern SM company, various hybrid schemes of AI techniques have also been applied. These combine the elements of learning, adaptation and evolution e.g. neuro-GA, grey-GA, neuro-fuzzy, etc. ([12]-[14]).

Although, there are many likely benefits of using more than one technique, a beforehand decision on which technique to apply for SM estimation, is nearly impossible. Often, adequate information of real life projects regarding size, maintenance history, human and management factors such as management focus, client attitude, the need for multi-location support teams etc. is unavailable, further complicating the objective estimation of SM effort. Till date no single estimation model has been successfully applied across a wide variety of projects.

In recent years, the Taguchi method (TM) has been increasingly used by companies for optimization of product and process performance [15]. However, most of the research based on TM has been limited to optimization of only a single response or quality attribute. In real life conditions though, the performance of a process/product demands multi-response optimization. Often engineering judgment is applied for process optimization, which is rather subjective and brings uncertainty into the decision making. This uncertainty can be overcome by using a hybrid TM and principal component analysis based approach [16].

Multivariate analysis is primarily used to reduce the data dimension and for a better understanding of the data by analyzing its covariance structure [17]. Principal component analysis uncovers unsuspected relationships, allowing the user to interpret the data in an alternate way. The goal of principal component analysis is to explain the maximum amount of variance with the fewest number of principal components. No research work has been reported so far on multi-variate analysis using PCA for SM effort estimation. Therefore, in the present paper, the TM coupled with PCA is applied to deal with the large number of effort data, in a more efficient manner.

3 Present Work

The objective of the present work is to apply the PCA technique to a large sized, commercial, real life dataset of SM from Syntel, an ISO 9001 cetified and NASDAQ listed application management and e-business solutions company. The open literature dataset of 14 effort drivers (Table 1) for 36 outsourced SM projects, estimated by 6 experts each is used (Appendix 1) [4]. This data is based on a legacy insurance system running on the IBM platform and coded predominantly in COBOL. The total size of the system was 1,386,852 lines and the number of programs were 1275. Further details of the syatem can be seen in [4].

Considering the number of independent parameters (effort drivers numbering 14), the fraction factorial design for conducting the experiment has been used. The number of levels for one of the independent parameters was fixed at 2 and for the rest 13 parameters, 3 levels were fixed. The Taguchi orthogonal arrays provide an efficient way of conducting minimum number of experiments that give full information of the factors that affect the responses, unlike traditional experimentation which considers only one factor at a time, while keeping the other parameters constant [15]. This technique was applied in [4] to limit the number of experiments from a maximum possible 3188646 (2^1 X 3^{13}) to a reasonable number (36) by using the standard $L_{36}(2^1$ X $3^{13})$ TOA. The choice of PCA as the empirical modeling tool was governed by the fact that too many predictors (14 here) are being considered, relative to the number of available observations (36).

Table 1. Effort drivers

Sl. No.	Effort Drivers
A	Existence of restart/recovery logic in batch programs
B	Percentage of the online programs to the total number of programs
C	Complexity of the file system being used
D	Average number of lines per program
E	Number of files (input, output, sort) used by the system
F	Number of database objects used by the system
G	Consistency and centralization of exceptional handling in programs
H	Whether structured programming concepts have been followed in the program
I	Percentage of commented lines of code to the total lines of code of the system
J	Number of programs executed as part of a batch job
K	Number of database structures used by a typical program
L	Percentage of the update programs to the total number of programs
M	Nature of service level agreement (SLA)
N	Whether structured programming concepts have been followed in the program

4 Principal Component Analysis

Principal component analysis (PCA) is a useful multivariate statistical method commonly used to reduce the data and avoid multicollinearity [17]. Principal Components (PC) form a smaller number of uncorrelated variables and are linear combinations (weighted average of the variables i.e. sum of the products between variables and the corresponding coefficients) of the original variables, that account for the variance in the data. For example, the present data of estimated SM effort is based on 14 drivers and our aim is to reduce the drivers (or variables) into a smaller number of PCs (or uncorrelated variables) for easier analysis of the data. The number of PCs can be determined using any one or combination of the methods given underneath [17]:

1. *Percent variance explained*: For example, the components that cumulatively explain 90-95% of the variance may be retained.
2. *Based on the size of eigen values*: According to the Kaiser criterion, the PCs with eigen values greater than 1 are retained.
3. *Analysis of Scree plot*.

Here, we have used the first method to find the number of PCs.

Before conducting PCA the distribution of data was checked. The histogram plot of Figure 1 shows a fairly normally distributed dataset, except the 2 outliers.

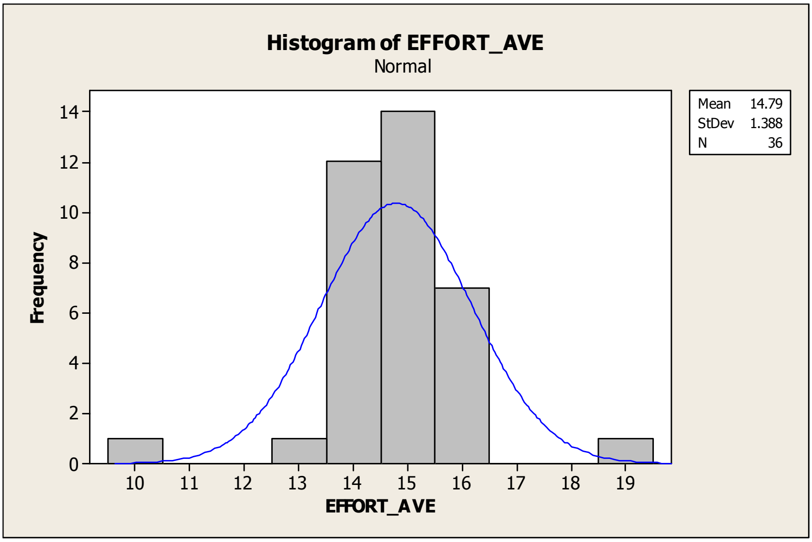


Fig. 1. Histogram showing normal distribution of data

Applying the Taguchi signal-to-noise ratio concept for the 'smaller-is-better' optimization criterion on the SM dataset, the PCA was conducted using the stepwise procedure as proposed by reference [16] and given below:

1. The S/N ratio of each expert's estimate obtained is normalized at first. The normalized array of m experts' estimated effort for n experimental runs can be represented by a matrix X^* as shown underneath:

$$x^* = \begin{pmatrix} x_1^*(1) & x_1^*(2) & \ldots & x_1^*(m) \\ x_2^*(1) & x_2^*(2) & \ldots & x_2^*(m) \\ \ldots & \ldots & \ldots & \ldots \\ x_n^*(1) & x_n^*(2) & \ldots & x_n^*(m) \end{pmatrix}$$

2. The correlation coefficient array (R_{ji}) corresponding to matrix x^* is computed as follows:

$$R_{ji} = \frac{\mathrm{cov}(x_i^*(j), x_i^*(l))}{\sigma_{x_i^*(j)} \times \sigma_{x_i^*(l)}}, j = 1, 2...., m, l = 1, 2...., m\,. \tag{1}$$

3. The eigen values and eigen vectors of matrix R_{ji} are computed as follows:

$$[R - \lambda(k)I_m)]v_k(j) = 0\,. \tag{2}$$

where $\lambda(k)$ is the *k*th eigen value and $v_k(j) = \left[v_{k1}, v_{k2}.....v_{kn}\right]^T$ are the eigen vectors corresponding to the eigen value $\lambda(k)$. The eigen values of the correlation matrix equal the variances of the principal components.

4. The principal component (PC) is computed as follows:

$$pc_i(k) = \sum_{j=1}^{n} x_i^*(j) \times v_k(j)\,. \tag{3}$$

where $pc_i(k)$ is the *k*th PC corresponding to the *i*th experimental run.

5. The total principal component index ($\hat{Pi}$) corresponding to the *i*th experimental run is computed as follows:

$$\hat{Pi} = \sum_{k=1}^{m} p_i(k) \times \lambda_i(k)\,. \tag{4}$$

where $\lambda_i(k) = \dfrac{\lambda(k)}{\sum_{k=1}^{m} \lambda(k)}$

5 Results and Discussion

The correlation coefficient array as shown in Table 2 is obtained using Eq. (1). Each PC has a corresponding eigen vector which is used to calculate the PC score (linear combinations of the data using the coefficients listed under PC1, PC2, and so on), which are comprised of coefficients corresponding to each variables. These coefficients indicate the relative weight of each variable in the component. The bigger the absolute value of the coefficient, the more important the corresponding variable is in constructing the PC. The computed eigen values using Eq. (2) alongwith their corresponding variance are given in Table 3, and the corresponding eigen vectors are given in Table 4.

Table 2. Correlation coefficients for the 6 experts' estimates

Correlation coefficient	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6
Exp 1	1.0000	0.6817	0.8937	0.5557	0.6230	0.7371
Exp 2	0.6817	1.0000	0.5880	0.5883	0.5522	0.6668
Exp 3	0.8937	0.5880	1.0000	0.5580	0.4797	0.7574
Exp 4	0.5557	0.5883	0.5580	1.0000	0.3011	0.4895
Exp 5	0.6230	0.5522	0.4797	0.3011	1.0000	0.5654
Exp 6	0.7371	0.6668	0.7574	0.4895	0.5654	1.0000

Table 3. Eigen values and variances

PC	Eigen values	% Variance	Cumulative variance
1	4.0524	67.54	67.54
2	0.7145	11.91	79.45
3	0.5326	8.87	88.32
4	0.3559	5.93	94.25
5	0.2669	4.44	98.69
6	0.0777	1.29	99.98

For the present problem as per Table 3, we can conclude that the first four principal components account for most of the variability (around 95%) in the data. The last two principal components account for a very small proportion of the variability and seem to be unimportant, and hence can be dropped in further analysis.

The first principal component PC1 with an eigen value greater than 1 alone represents a variance equal to 67.54% of the total variability, suggesting that this is the most important PC. This is followed by PC2 which in comparison to PC1 represents a significantly lower variance of 11.91% of the total variability, suggesting that this PC is much less important than PC1.

Table 4. Eigen vectors for 6 experts' estimates

Eigen vectors	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6
Exp 1	0.699	-0.142	-0.661	0.032	-0.154	0.164
Exp 2	-0.406	-0.452	-0.174	0.272	0.141	0.710
Exp 3	-0.206	0.605	-0.150	-0.399	-0.433	0.469
Exp 4	-0.296	0.483	-0.556	0.365	0.430	-0.223
Exp 5	0.063	-0.070	-0.084	-0.715	0.678	0.109
Exp 6	0.458	0.411	-0.084	0.346	0.350	0.430

Table 5. Principal component and TPCI values

Run	PC1	PC2	PC3	PC4	PC5	PC6	TPCI
1.	0	0	0	0	0	0	0.00
2.	0.18	0.38	-0.80	-0.36	0.22	0.81	0.51
3.	0.31	0.83	-1.71	-0.10	1.01	1.66	1.17
4.	0.23	0.27	-0.83	-0.39	0.50	0.90	0.60
5.	0.09	0.42	-0.58	0.03	-0.05	0.56	0.36
6.	0.09	0.23	-0.94	-0.23	0.56	1.05	0.71
7.	0.33	0.42	-0.84	-0.23	0.47	1.03	0.70
8.	0.25	0.39	-0.79	-0.22	0.24	0.81	0.53
9.	0.21	0.46	-0.87	-0.20	0.34	0.78	0.52
10.	0.23	0.29	-0.78	-0.30	0.32	0.93	0.61
11.	0.29	0.34	-0.79	-0.28	0.38	0.99	0.66
12.	0.21	0.46	-0.86	-0.12	0.26	0.77	0.51
13.	0.25	0.39	-0.78	-0.18	0.21	0.81	0.53
14.	0.26	0.39	-0.89	-0.42	0.62	0.98	0.67
15.	0.07	0.34	-0.62	-0.11	0.16	0.87	0.57
16.	0.19	0.34	-0.80	-0.10	0.18	0.89	0.58
17.	0.13	0.32	-0.94	-0.34	0.57	0.99	0.67
18.	0.21	0.31	-0.73	-0.27	0.15	0.77	0.49
19.	0.29	0.50	-0.86	-0.33	0.44	0.90	0.61
20.	0.21	0.48	-0.92	-0.22	0.44	0.97	0.65
21.	0.32	0.63	-0.96	-0.23	0.55	1.12	0.78
22.	0.32	0.45	-0.87	-0.37	0.55	1.07	0.73
23.	0.15	0.58	-1.05	-0.20	0.58	0.93	0.64
24.	0.19	0.51	-0.89	0.06	0.18	0.92	0.62
25.	0.18	0.66	-1.02	-0.23	0.52	0.66	0.45
26.	0.07	0.44	-1.09	-0.19	0.71	1.10	0.76
27.	-0.03	0.58	-1.16	0.32	0.39	0.98	0.70
28.	-0.05	0.49	-1.20	-0.14	0.75	0.99	0.70
29.	0.09	0.51	-0.94	0.13	0.16	0.81	0.55
30.	0.21	0.48	-0.97	-0.28	0.58	0.95	0.65
31.	0.14	0.59	-0.97	0.10	0.23	0.90	0.62
32.	0.30	0.49	-0.82	-0.29	0.32	0.92	0.61
33.	0.38	0.43	-1.15	-0.31	0.75	1.33	0.92

Table 5. (*continued*)

34.	0.22	0.75	-1.03	-0.14	0.47	0.73	0.51
35.	0.26	0.55	-0.95	0.01	0.42	1.15	0.80
36.	0.11	0.52	-1.11	-0.16	0.72	1.18	0.83
						Avg. TPCI	0.63

Table 6. Response table based on S/N ratio of TPCI

Level	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	4.32	4.43	4.46	3.91	4.19	4.70	3.24	3.61	3.62	4.12	3.77	4.53	4.31	5.09
2	3.75	4.10	4.35	4.19	4.49	4.27	4.15	4.09	4.20	4.31	4.51	4.27	4.37	4.13
3	--	3.58	3.30	3.97	3.42	3.17	4.62	4.35	4.22	3.65	3.77	3.33	3.36	2.95
Delta	0.57	0.85	1.15	0.28	1.07	1.52	1.37	0.73	0.59	0.66	0.74	1.20	1.01	2.14
Rank	13	8	5	14	6	2	3	10	12	11	9	4	7	1

The six principal components PC1 to PC6 and their total principal component index (TPCI) for all the 36 experimental runs are computed using Eqs. (3) and (4) and tabulated in Table 5. All the computation and analysis is performed using the Minitab software [17].

The S/N ratios obtained from TPCI values for all the experimental runs are used to find out the ranking of all the 14 factors by finding out the Delta statistic at all factor levels, as shown in the response table (Table 6). The parameter N (whether structured programming concepts have been followed in the program) is found to have a considerably dominant effect on the effort and is ranked at number 1 while the parameter D (average number of lines per program) has the least significant effect and is ranked last at number 14.

This result is different from that of the single response optimization problem, as obtained by ranking the drivers based on the S/N ratio obtained, from the calculated mean of the predicted effort of 6 experts (Table 7). Though the parameter N continues to have the most dominant effect on the effort and is ranked at no. 1 but now the

Table 7. Response table based on S/N ratio of Means

Level	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	0.59	0.57	0.55	0.59	0.56	0.54	0.63	0.61	0.61	0.58	0.60	0.56	0.56	0.51
2	0.65	0.62	0.61	0.63	0.61	0.62	0.62	0.63	0.62	0.61	0.59	0.61	0.61	0.63
3	--	0.67	0.70	0.65	0.69	0.70	0.61	0.63	0.64	0.67	0.67	0.69	0.69	0.72
Delta	0.06	0.10	0.15	0.06	0.12	0.16	0.02	0.02	0.02	0.09	0.07	0.13	0.13	0.21
Rank	10	7	3	11	6	2	14	13	12	8	9	4	5	1

parameter G (consistency and centralization of exceptional handling in programs) has the least significant effect and is ranked last at number 14. Further details of the above approach are presented in reference [18]. As per expectation, the obtained results of driver ratings are slightly different in both the cases, as shown in Table 8, because of the varying influence of 6 different experts' effort prediction in case of simultaneous

Table 8. Comparison of ranking of effort drivers based on TPCI and Mean Effort

Sl. No.	Driver	Ranking based on TPCI	Ranking based on Mean Effort
1	A	13	10
2	B	8	7
3	C	5	3
4	D	14	11
5	E	6	2
6	F	2	6
7	G	3	14
8	H	10	13
9	I	12	12
10	J	11	8
11	K	9	9
12	L	4	4
13	M	7	5
14	N	1	1

Table 9. Analysis of Variance

Predictor	Coef	SE Coef	T	P
Constant	-0.74444	0.06145	-12.11	0.000
A	0.06556	0.01310	5.00	0.000
B	0.050833	0.008023	6.34	0.000
C	0.076250	0.008023	9.50	0.000
D	0.030417	0.008023	3.79	0.001
E	0.062083	0.008023	7.74	0.000
F	0.083333	0.008023	10.39	0.000
G	-0.011667	0.008023	-1.45	0.161
H	0.010833	0.008023	1.35	0.191
I	0.013333	0.008023	1.66	0.111
J	0.047917	0.008023	5.97	0.000
K	0.032083	0.008023	4.00	0.001
L	0.068750	0.008023	8.57	0.000
M	0.065417	0.008023	8.15	0.000
N	0.106250	0.008023	13.24	0.000
	S = 0.0393062	R-Sq = 97.1%	R-Sq(adj) = 95.2%	

estimation, as against single response optimization problem based on the mean effort of 6 experts.

Source	DF	SS	MS	F	P
Regression	14	1.098844	0.078489	50.80	0.000
Residual Error	21	0.032444	0.001545		
Total	35	1.131289			

ANOVA (Analysis of variance) analysis was conducted on the given data and a linear regression model (Eq. 5) was fitted for evaluating the TPCI in terms of various parameters. The parameters L, M, N etc. (higher value of T or lower value of P) are found to have a dominant effect on the effort, while the parameters G, H, I etc. (lower value of T or higher value of P) have a less significant effect. An excellent agreement ($R^2 = 0.97$) is observed between the linear model predicted values and experimental data indicating the consistency of data and the goodness of fit of the linear model (Table 9). However, the same may not be true beyond the present range of parameters and more so when there is a non-linear relationship and an interaction between the effort drivers and response.

$$\begin{aligned} TPCI = & - 0.744 + 0.0656\ A + 0.0508\ B + 0.0762\ C + 0.0304\ D + 0.0621\ E + \\ & 0.0833\ F - 0.0117\ G + 0.0108\ H + 0.0133\ J + 0.0479\ K + 0.0321\ L + \\ & 0.0688\ M + 0.0654\ N + 0.106\ O\ . \end{aligned} \tag{5}$$

6 Conclusions

In this paper, the effectiveness of the PCA approach in empirical modeling of effort estimation in outsourced software maintenance is presented. Since the effort is dependent on a large number and variety of drivers, and besides their relationship is often quite complex, PCA can be used as an effective approach to reduce the data and predict the variance in SM effort estimation. However, PCA can also be used further to determine the number of factors to be extracted for a factor analytic study of software development and maintenance projects. This study may then be used to describe the covariance among variables in terms of a few underlying factors. As a future course of action the proposed approach can be evaluated against unseen input data of a variety of projects.

References

1. IEEE Standard 1219: Standard for Software Maintenance. IEEE Computer Society Press, Los Alamitos (1998)
2. Boehm, B., Abts, C., Chulani, S.: Software development cost estimation approaches – a survey. Ann. Softw. Eng 10, 177–205 (2000)
3. Shukla, R., Misra, A.K.: AI Based Framework for Dynamic Modeling of Software Maintenance Effort Estimation. In: International Conference on Computer and Automation Engineering, pp. 313–317 (2009)

4. Rao, B.S., Sarda, N.L.: Effort drivers in maintenance outsourcing - an experiment using Taguchi's methodology. In: Seventh IEEE European Conference on Software Maintenance and Reengineering, pp. 1–10 (2003)
5. Ahn, Y., Suh, J., Kim, S., Kim, H.: The software maintenance project effort estimation model based on function points. J. Softw. Maint. Evol.: Res. and Pract. 15(2), 71–78 (2003)
6. Bhatt, P., Shroff, G., Anantram, C., Misra, A.K.: An influence model for factors in outsourced software maintenance. J. Softw. Maint. Evol.: Res. and Pract. 18, 385–423 (2006)
7. Bhatt, P., Williams, K., Shroff, G., Misra, A.K.: Influencing factors in outsourced software maintenance. ACM SIGSOFT Softw. Eng. Notes 31(3), 1–6 (2006)
8. Jorgensen, M.: Experience with accuracy of software maintenance task effort prediction models. IEEE Trans. Softw. Eng., 674–681 (1995)
9. Martín, C.L., Márquez, C.Y., Tornés, A.G.: Predictive accuracy comparison of fuzzy models for software development effort of small programs. J. Syst. Softw. 81(6), 949–960 (2008)
10. Srinivasan, K., Fisher, D.: Machine learning approaches to estimating software development effort. IEEE Trans. Softw. Eng. 21(2), 126–137 (1995)
11. Grimstad, S., Jørgensen, M.: Inconsistency of expert judgment-based estimates of software development effort. J. Syst. Softw. 80(11), 1770–1777 (2007)
12. Shukla, R., Misra, A.K.: Estimating Software Maintenance Effort - A Neural Network Approach. In: 1st India Software Engineering Conference, Hyderabad, pp. 107–112. ACM Digital Library (2008)
13. Pendharkar, P.C., Subramanian, G.H., Rodger, J.A.: A probabilistic model for predicting software development effort. IEEE Trans. Softw. Eng. 31(7), 615–624 (2005)
14. Shukla, K.K.: Neuro-genetic prediction of software development effort. Inform. Softw. Tech. 42, 701–713 (2000)
15. Phadke, M.S.: Quality Engineering Using Robust Design. Prentice-Hall, Englewood cliffs (1989)
16. Fung, C.P., Kang, P.C.: Multi-response optimization in friction properties of PBT composites using Taguchi method and principal component analyses. J. Mater. Proc. Tech. 170, 602–610 (2005)
17. Minitab, `http://www.minitab.com`
18. Shukla, R., Misra, A.K.: Software Maintenance Effort Estimation - Neural Network Vs Regression Modeling Approach. Int. J. Futur. Comp. Applic. (Accepted, 2010)

Appendix 1.

Sl. No.	A	B	C, D, … K, L	M	N	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Mean effort
1.	1	10		1	1	12	8	10	6	12.67	10	9.8
2.	1	10		5	3	15	11	19.3	10	17.44	12	14.1
3.	1	10		10	5	17	16	22.9	20	21.53	18	19.2
4.	1	10		10	5	15	12	17.4	10	19.44	13	14.5
5.	1	10		1	1	14	10	17.4	10	12.86	12	12.7
6.	1	10		5	3	15	14	17.4	12	19	13	15.1
7.	2	10		5	5	15	12	18.2	10	18.23	15	14.7
8.	2	10		10	1	15	11	18.2	10	16.66	13	14.0
9.	2	10		1	3	15	11	18.2	12	17.1	13	14.4
10.	2	10		10	3	15	12	18.2	9	17.78	13	14.2
11.	2	10		1	5	15	12	18.2	9	18.01	14	14.4
12.	2	10		5	1	15	11	18.2	12	16.06	13	14.2
13.	1	30		1	3	15	11	18.2	10	16.22	13	13.9
14.	1	30		5	5	15	12	18.2	11	20.37	14	15.1
15.	1	30		10	1	14	12	18.2	9	15.36	13	13.6
16.	1	30		5	1	15	12	18.2	10	15.6	13	14.0
17.	1	30		10	3	15	13	18.2	12	19.67	13	15.1
18.	1	30		1	5	15	11	18.2	9	16.43	12	13.6
19.	2	30		5	5	15	11	19	11	18.59	14	14.8
20.	2	30		10	1	15	12	19	12	17.92	14	15.0
21.	2	30		1	3	15	12	20.1	12	18.72	16	15.6
22.	2	30		10	3	15	12	19	10	19.68	15	15.1
23.	2	30		1	5	15	12	19	15	18.64	14	15.6
24.	2	30		5	1	15	12	19	12	14.51	14	14.4
25.	1	50		5	3	15	10	18.5	16	18.17	13	15.1
26.	1	50		10	5	15	14	18.5	15	19.61	14	16.0
27.	1	50		1	1	15	14	18.5	18	14.24	14	15.6
28.	1	50		1	5	15	14	18.5	18	19.43	13	16.3
29.	1	50		5	1	15	12	18.5	14	13.89	13	14.4
30.	1	50		10	3	15	12	18.5	13	19.21	14	15.3
31.	2	50		1	1	15	12	19.3	14	14.53	14	14.8
32.	2	50		5	3	15	11	19.3	10	17.58	14	14.5
33.	2	50		10	5	16	14	19.3	12	20.95	16	16.4
34.	2	50		10	1	15	10	19.3	16	17.28	14	15.3
35.	2	50		1	3	15	13	19.3	12	16.43	16	15.3
36.	2	50		5	5	15	14	19.3	15	19.5	15	16.3

Process Model Based Incremental Project Planning

Edward Fischer

Clausthal University of Technology, Institute of Computer Sciences,
Julius-Albert-Str.4, 38678 Clausthal-Zellerfeld, Germany,
ef@tu-clausthal.de

Abstract. It is widely accepted that process models can significantly increase the likelihood of a project to finish successfully. A very basic aspect of actually using a process model is to derive a project plan thereof and keep it up to date. However, this is a tedious task – especially when each document might undergo numerous revisions. Thus, automation is needed. However, current approaches based on workflows or change management systems do not provide incremental update mechanisms: process engineers have to define them by themselves – especially when they develop an organization specific process model. On the other hand, incremental model transformations, known from the model driven development domain, are to low-level to be of practical use. In fact, proper high-level model transformation languages are yet subject to research. In this paper we present a process language which integrates both: process modeling languages and incremental model transformations.

Keywords: Project Planning, Process Models, Incremental Transformation.

1 Introduction

Motivation. Process models define who has to do what and when in a project. Thus they help a project leader to remember important steps to be done. Of course, they cannot give the exact number of documents to be produced for a specific project. Instead, they abstract from concrete documents by just giving document types (e.g. "architecture", "test specification" "test protocol" etc) and rules for instantiation (e.g. "for each critical unit, a "test specification" has to be produced"). A project leader can apply these planning rules on his specific project to derive a concrete project plan, which covers the complete list of documents to be produced. Afterwards, he can enrich his project plan with further information like task assignments (which person has to create which scheduled document) and completion state (which planned documents have been finished?).

Software development projects are highly dynamic. Documents, once created, are likely to be updated later on. This also implies the need to update the project plan. For instance, consider an architecture document in which the three critical units A, B and C were identified at some time point t1. Following the process model planning rules, a project leader had to schedule three respective specification documents. At some later point in time, say t2, the architecture document was revised. Thereby, two additional units D and E were added to the list of cirtical units while C was removed

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 285–300, 2010.

thereof. Now, the project leader has to update his project plan in such a way, that the specification document for C is removed and two specification documents for D and E are added. However, those two specification documents for A and B have to be kept untouched during the project plan update. This is crucial for not loosing additional information like task assignment and completion states.

Problem to be solved. Applying planning rules by hand to either create or update a project plan is a tedious task. So it should be automated. Therefore an executable language is needed to formalize those planning rules. However, that language has to fulfill two requirements for being practically useful:

I. The language should be easy to understand by a process engineer (who has the job to encode the process model in question with its respective planning rules).
II. The process engineer should not care about the update problem. In other words, there should be an algorithm which can update the generation of a project plan for *any* process model with *any* set of planning rules defined in that language. With that fixed algorithm, a process engineer does not need to define update mechanisms himself.

Why the problem has not been solved yet. Process modeling languages are good for being used by process engineers to define process models on a high level (thus solving issue I). However, they lack of incrementally updatable constructs (missing issue II). On the other hand, current incremental model transformations [8] solve issue II but not issue I. The most advanced approaches from that field are those based on Triple Graph Grammars [9]. Thereby, models have to be encoded as graphs and the transformation has to be encoded through graph grammars rules. This, however, is too low-level for being of practical use (see e.g. [10]). Although appropriate languages are subject to research (see e.g. [10, 11, 12]) – there is no approach dealing with the domain of process models. As a result these approaches fail to address issue I.

Contribution of this Paper. The basic idea to construct a language meeting the solving the issues I and II is to integrate the incremental model transformation approach (solving issue II but not I) with a process modeling language (solving issue I but not II).

Concerning issue II, we regard planning rules as a transformation from the artifact library[1] to the project plan (see fig. 1). That transformation is supposed to be executed in an incremental way. That is, instead of recreating the project plan from scratch, each transformation execution (1) will just update the present project plan keeping those parts untouched which are already consistent with current project state (2).

Concerning issue I, this paper modifies the process language *V-Modell XT Metamodell* (VM³), version 1.3 [18, 19, 20], by replacing rather informal dependency constructs with more formal and incrementally executable planning rules. The choice of VM³ is arbitrary for the presented approach of this work. Any other process modeling language (e.g. *SPEM* [21]) would be as likely usable. However, VM³ was used to define the process model *V-Modell XT* [22], which in turn is currently

[1] An artifact library is the set of all documents and versions thereof in a project.

used in a customized version by EADS (European Aeronautic Defence and Space Company) [23]. As a part of the aerospace industry, EADS needs rigorous process models with appropriate planning support. This, in turn, shows the practical relevance of the proposed approach.

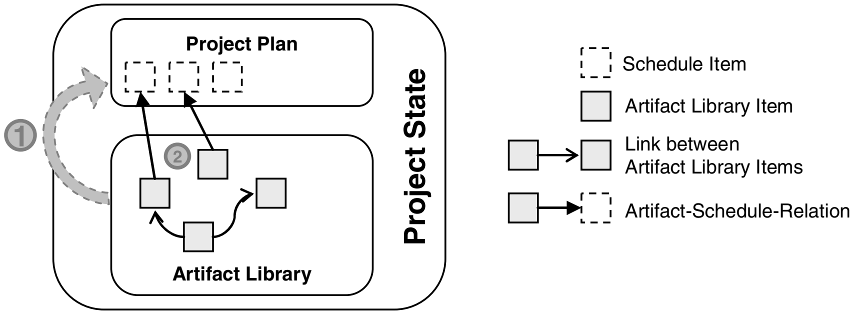


Fig. 1. Project planning as an incremental transformation

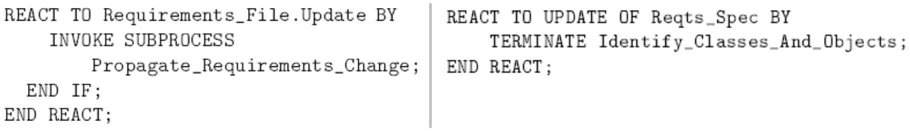
Structure of this Paper. In section 2, the state of the art is revisited focusing on process languages and incremental transformations. Section 3 presents the extensions made to the process language VM³. Finally, section 4 covers the evaluation and section 5 concludes this paper.

2 State of The Art

There are two general directions relevant to this work: The first one, discussed in section 2.1, is about process modeling languages (or so called software process metamodels). The second one, discussed in section 2.2, is about incremental transformations (which on their own have nothing special to do with the domain of process models).

2.1 Process Modeling Languages

JIL [17] subsumes a wide range of process languages like EPOS [24], Process Weaver [25], Merlin [26], Melmac [27], Marvel [28] (just to name a few) and its paradigms. That is, JIL features proactive and reactive controls, exception handling, flexible control flow, pre- and post conditions, manually and automatically executed steps (typical for workflow languages like XPDL [29]), hierarchies and others. Although all of these paradigms help to define executable process models on a high level, there is an important draw back when it comes to changes. In JIL (and any other current process languages), changes will be propagated through events. Reactions (and exception handlers as a special case thereof) can be defined to dispatch those events to perform necessary updates. Two examples of those reaction definitions are shown in fig.2 (using JIL's concrete syntax). These examples were taken from [16] and [17]. Please note the original (thus here cited) explanations: the process engineer has to define appropriate reactions himself. To state that more precisely, *Propagate_Requirements_Change* is not a build-in function of JIL. This is a user-defined function (i.e. sub process) which has to be implemented by the process engineer. Considering the second example, terminating a scheduled item would lead to a loss of any information attached to it meanwhile.



```
REACT TO Requirements_File.Update BY
    INVOKE SUBPROCESS
        Propagate_Requirements_Change;
  END IF;
END REACT;
```

```
REACT TO UPDATE OF Reqts_Spec BY
    TERMINATE Identify_Classes_And_Objects;
END REACT;
```

Fig. 2. Process modeling language JIL and its way to deal with changes

2.2 Incremental Transformations

Straightforward Implementation. The first idea which comes to mind is to use any programming language to implement planning rules as a transformation which takes the set of all current documents as input and creates a project plan as output. However, this approach recreates the project plan on each execution. Thereby, any additional information added to the created plan (task assignments, completion state etc) would be lost. In addition, ordinary programming languages are too low level for process engineers. As a result, this approach fails to address both issues II and I.

Concurrent Versioning Systems. Considering just issue II, a popular approach is to suggest concurrent versioning systems (like CVS or Subversion) to merge the old project plan with the newly generated one. However, merging different versions of a generated project plan with CVS would be similar to the situation of merging binaries of a program rather than its source code. Merging by line similarity [1] neither work out for binaries nor for any other generated artifacts. To state that in a more formal way, the problem of line similarity merging is that the nature of the transformation and the formal structure of the resulting artifacts are ignored [2, 3]. As a result, these approaches fail to address issue II.

Model Merging. Model merging improves the approach of merging by similar lines through either merging by structure similarity [4, 5] or merging by UUIDs (universally unique identifiers) [6, 7]. However, using UUIDs will not work out as each transformation execution will create a new project plan version with completely new UUIDs. Thus, no elements can be matched between two subsequently generated project plan versions. Using structural similarity, on the other hand, will often lead to wrong matches since a project plan has a lot of similar scheduling entries. However, it is crucial to map task assignments and completions states correctly. As a result, these approaches still fail to address issue II.

Triple Graph Grammars. Triple graph grammars (TGG) were introduced by Schürr in [9]. Fig. 3 shows the general idea. Grammars GGL and GGR are used as modeling languages for the Graphs GL and GR. Now, a third Grammar GGC is introduced which subsumes both Grammars GGL and GGR (i.e., GGC. "knows" all terminal types of both other Grammars) while also providing some terminal symbols of its own, which are used to form the elements in the correspondence Graph GC. Based on that setting, the following application scenarios can be defined: First, if there is just GL but neither GC nor GR, a *straightforward model transformation* can be performed. Thereby, rules defined by GGC are applied to GL, producing elements for GC and GR. The application itself is based on pattern matching using graph morphisms. Due to the undirected rule definition, the transformation is also possible for the opposite direction, i.e. GR is present with GC and GL missing. As a second application scenario, consider having all three Graphs GR, GC and GL available, in example after performing a straight forward model transformation. We can now change one Graph, for example GR, and update both others by an *incremental model transformation* [8, 31]. As an effect, GR (and GC) will not be completely dismissed and completely created from scratch again; just the changes implied by the modification of GL will be propagated to GR.

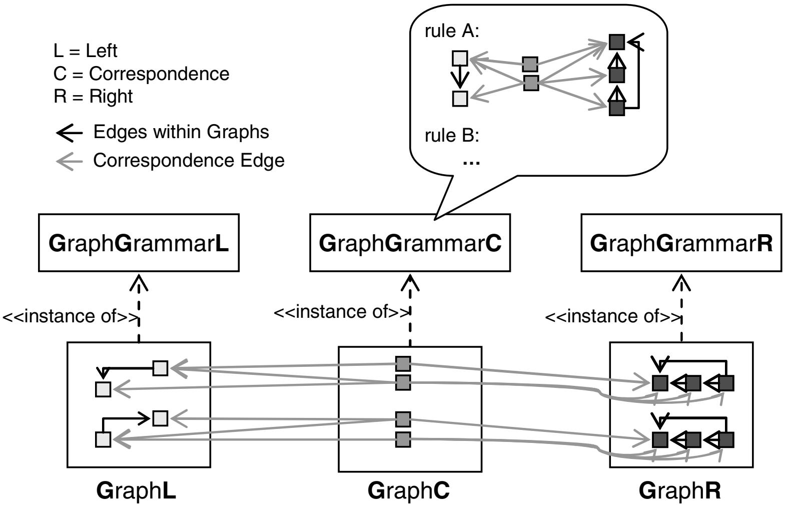


Fig. 3. Triple Graph Grammars

Although this concept looks promising at first glance, it has a serious shortcoming: It requires transformations to be defined through *graph rewriting rules* and *graphs* as the underlying data structure. Current research activities show, however, that today's graph transformation languages lack of practical expressiveness (see e.g. [10]). Thus, appropriate languages are yet subject to research: There are a number of miscellaneous approaches which are too specific to the transformations used (e.g. [11]), or to specific to the revisions allowed (e.g. [12]).

Workflows, Change Management and Others. Vendors of Workflows and Change Management Systems like ClearQuest [13], Notes [14] or Jazz [15] claim that they can deal with updates. Their main argument is that their process modeling language is expressive enough to express any change reaction to perform the desired update. However, this argument is true for ordinary programming languages, too. So just the ability to integrate any algorithm to update a project plan does not solve the actual problem of how that algorithm should look like. In fact, even Osterweil et al., creators of JIL, explicitly admitted in [16] and [17] that the process engineer still has to define update reactions himself. Thus, current process modeling languages (including any workflow or change management systems based upon) do not provide automatic update resolutions – especially when conflicts arise. As a result, these approaches fail to address issue II.

3 Solution: Updateable Process Language

The solution proposed in this paper is an integration of triple graph grammars with an existing process modelling language. Section 3.1 will cover the most important parts of the abstract syntax and semantics of the proposed language. As there is not enough room to explain every construct in full detail, just a brief overview is given. Section 3.2 will cover the incremental update mechanism, showing how process models and planning rules defined with that language can be incrementally updated.

3.1 Syntax and Semantics

Fig. 4 and fig. 5 present the abstract syntax of the proposed language using UML Class Diagrams. The entities with light grey background resemble document types and their possible contents, as well as their relation to other process model concepts like roles and phases. We omit activities as they are not important to this work (or, if you like, imagine that activity instances are scheduling items in the generated project plan). The entities with dark grey background make up the formal planning rule structure.

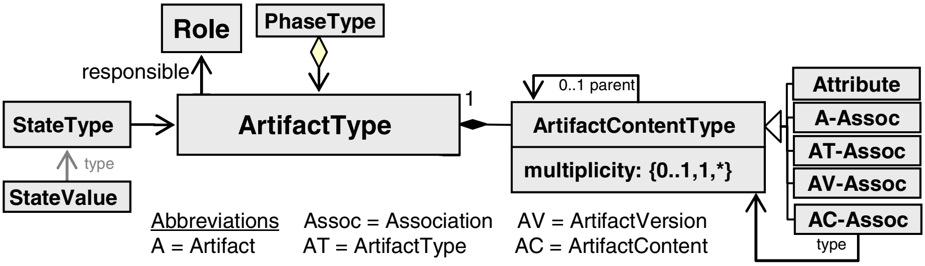


Fig. 4. Abstract syntax of basic data structures

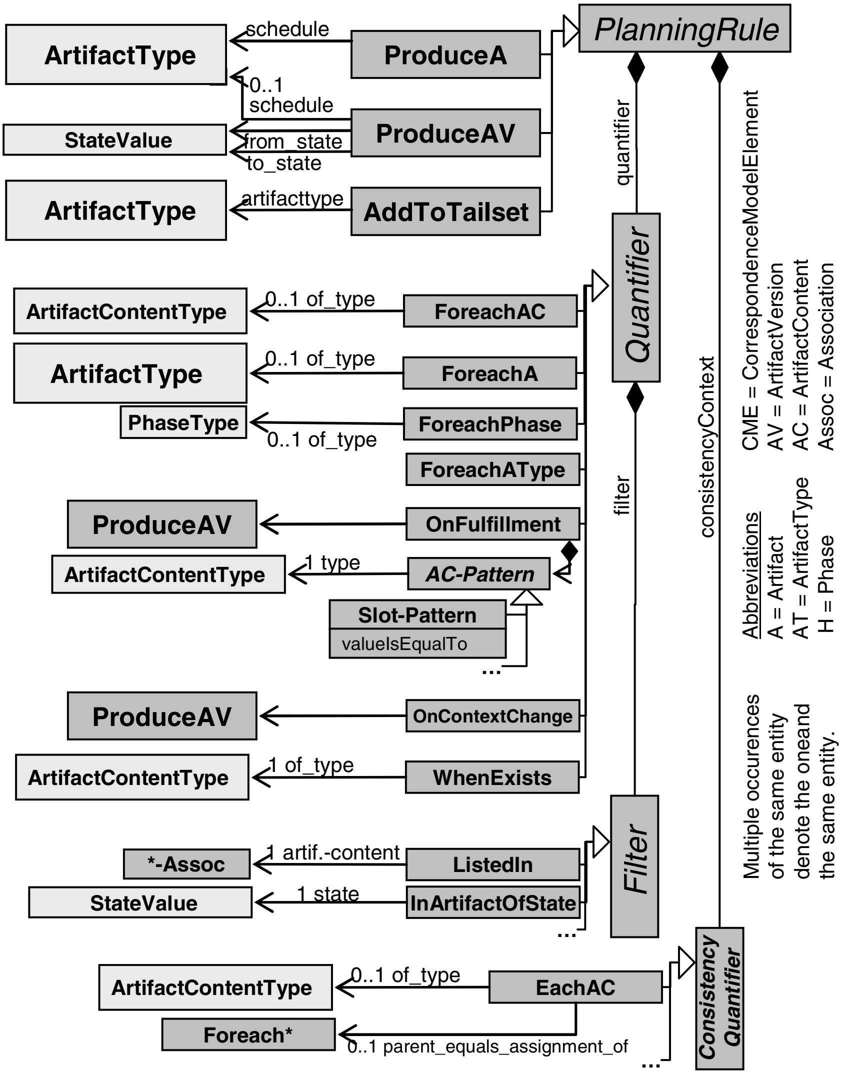


Fig. 5. Abstract syntax of planning rules (dark grey)

ProduceA. A *Produce-A*rtifact planning rule should be used to express when a new artifact should be scheduled. See Fig.6 for an example. ProduceA-Rules may use any quantifiers and any filter (we have shown only two here). The concept of quantifiers used here is basically the same as known from predicate logic. However, there are two levels of typing added here: the first one is the quantifier's name (e.g., the domain of a ForeachAC-Quantifier is the set of all artifact contents only). The second level is a logical type which is optional and can be used by that of_type-assocication. This way, only specific artifact contents can be quantified (see fig.5 again). Filters provide a mean to restrict quantifiers by relations between different artifacts or artifact contents – which are similar to predicates in predicate logic.

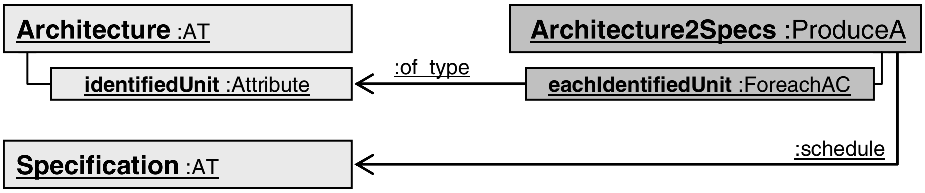

Architecture :AT
identifiedUnit :Attribute
:of_type
Architecture2Specs :ProduceA
eachIdentifiedUnit :ForeachAC
:schedule
Specification :AT

Fig. 6. Example of a rule defined using the proposed language. That rule states the following: "Schedule a *Specification document* for each identified unit in an *Architecture* document".

ProduceAV. A *Produce*-*A*rtifact-*V*ersion planning rule should be used to implement simple artifact state models. A classical example, as shown in fig. 7, is the quality assurance state machine featuring an *in process*, *submitted* and *finished* state with appropriate state transitions. A transition from *in process* to *submitted* might require a review protocol to be scheduled. This can be declared the same way as with a ProduceA-Rule. A transition from *submitted* to *finished* might require that protocol to contain a positive evaluation result. This can be expressed using the OnFulfillment-Quantifier. The purpose of that quantifier is to find that particular ("latest") protocol which is relevant for the current state transition. By adding content patterns, we can restrict that transition to trigger only if that document has satisfying contents. For example, if the evaluation result is negative, the transition to *finished* will not trigger. However, another transition might trigger than – e.g. one defined from *submitted* to *in process*, requiring a negative review result.

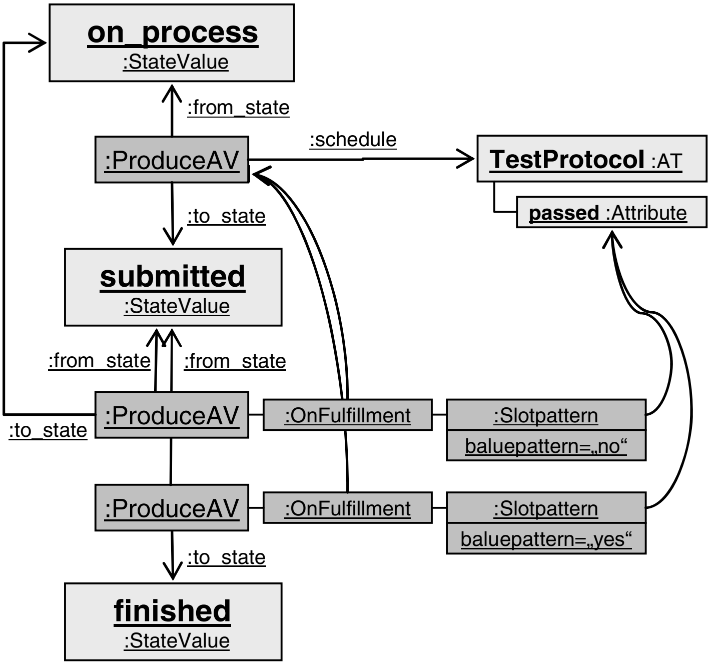


Fig. 7. Example of three artefact version rules making up a simple document state model

AddToTailset. An AddToTailoringset planning rule should be used to encode dynamic tailoring. The execution of tailoring rules will not create scheduling items, though. Instead, it will control the relevant process model potion (referred to as the tailoring set). This in turn directly influences the scope of the ForeachAT-Quantifier and respective rules defined upon. A simple example is shown in fig. 8: the artifact type HW Specification is added to the tailoring set when there is at least one hardware unit among the identified in any architecture document. As this example makes clear, tailoring rules are heavily based on the WhenExists-Quantifier so that using other quantifiers makes no sense. However, this and similar restriction are not shown in fig.5 to make the overall structure of rules (rules, quantifiers, filters) more convenient.

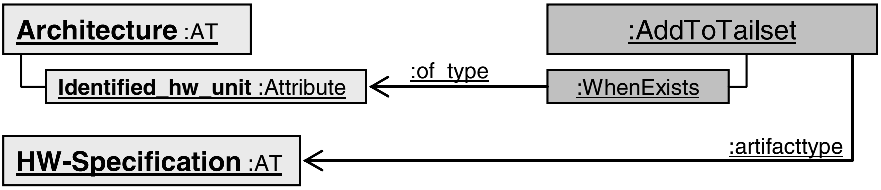


Fig. 8. Example of a tailoring rule usage

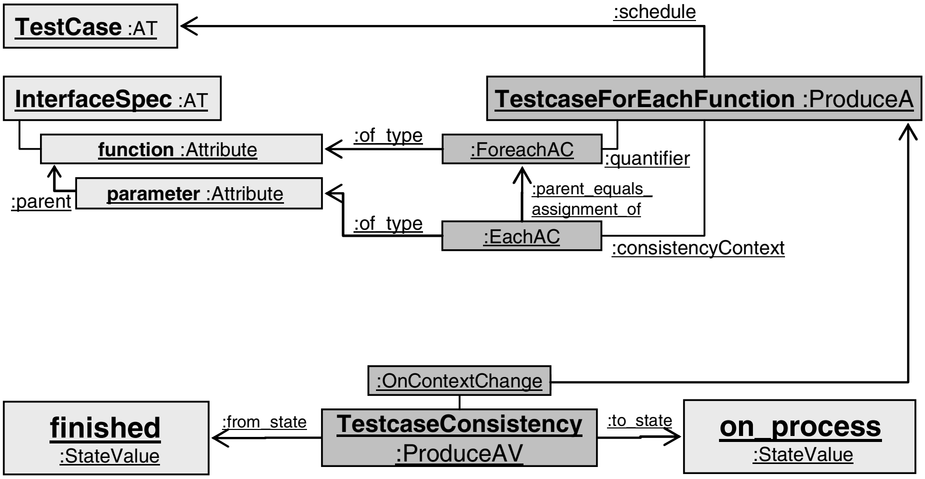


Fig. 9. Example of a consistency context usage

ConsistencyContext. A consistency context can be optionally added to a planning rule to trigger rules by *changes* in the artifact library (OnContextChange-Quantifier) rather than the pure *existence* of some elements there in. Consider fig. 9 for an example. In the upper half, a ProduceA-Rule named "TestcaseForEachFunction" is defined which schedules a *TestCase* document for each identified *function* in any *InterfaceSpec* document. Please note that identified parameters of a function are not used by that rule for producing new test cases. In other words, there is always one test case per function in this example to keep things simple. Now consider the consistency context definition which is added by extending the consicencyContext aggregation with an *EachA*rtifact*C*ontent-Quantifier. This way, the consistency context of each rule application is defined as the set of all elements matching the EachAC-Quantifier. In this case, this is the set of all parameters having the function in question as their parent. Any time this set changes the consistency context is regarded as having changed, too. For the example of fig. 8 this is the case when a new parameter is added or removed from a function. Please note that in such a case nothing would happen to TestCase documents produced. The lower half of fig.8 describes a second rule which defines a reaction on that consistency context change. The ProduceAV rule schedules a new version of the affected TestCase document by using the OnContextChange-Quantifier.

3.2 Incremental Update Mechanism

Fig.10 shows how the incremental transformation approach is integrated with the language presented in section 3.1. The basic idea is to store instances of planning rule applications (which are called *PlanningRuleInstances* here). Each of those instances stores the values which were assigned to that rule's quantifiers (*QuantifierAssignmentCombinations, or QAC for short*). Note that a rule may be applied multiple times leading to multiple rule instances with different *QACs*. In addition, a PlanningRuleInstance stores a link to the *SchedulingItem* it has produced. Now, on any subsequent transformation, PlanningRuleInstances are compared by their *QACs*. If a new combination occurs, it's *ScheduleItem* is added to the project plan. If a combination does not longer exist, the *ScheduleItem* it has produced is removed.

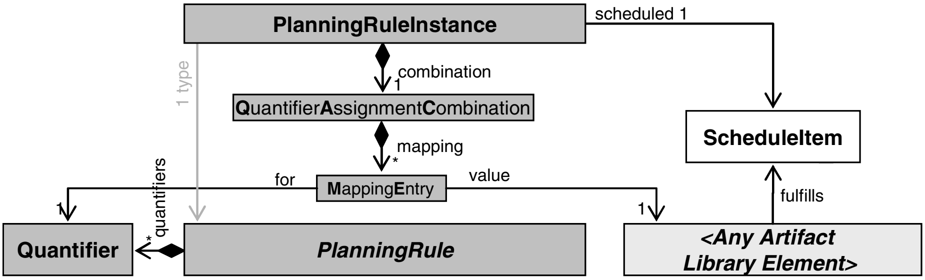


Fig. 10. Abstract structure for the incremental update mechanism

Yet, this is not the full story as those add- and remove actions depend on the kind of planning rule (i.e. A-Rule, AV-Rule or Tailoring-Rule). However, to close this section, the relation to triple graph grammars is made more explicit: The set of all *PlanningRuleInstances* (together with their *QAC*s) resembles exactly the correspondence model between the artifact library as the one graph, and the set of scheduling items as the other. Fig. 11 depicts the structure of the artefact library in relation to the process model structure.

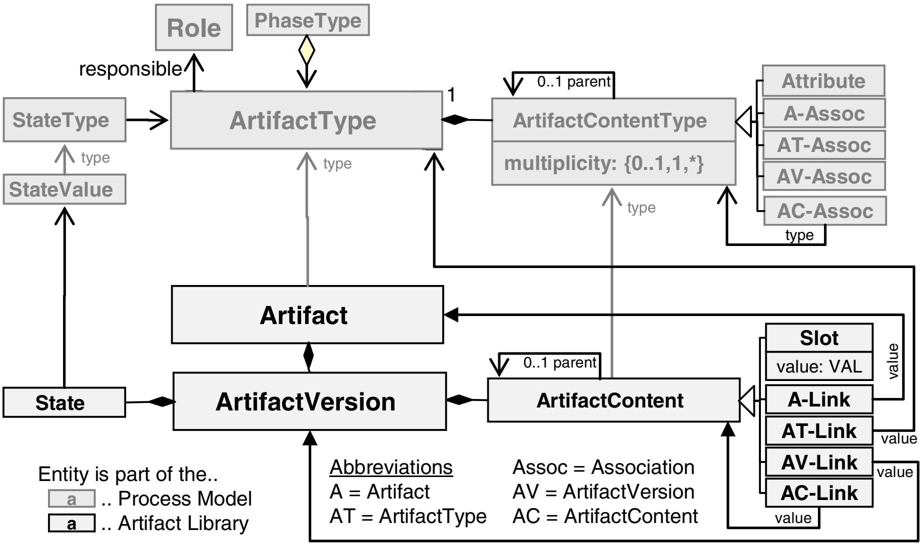


Fig. 11. Abstract structure of the artifact library

4 Evaluation

Method. For evaluation, the process model V-Modell XT [22] was considered. This process model, slightly changed, is in fact used in practice by EADS [23]. Five cases of typical planning problems EADS project manager would have were outlined. Until now, respective planning rules were informally defined in the process model. Thus, adopting them was a manual, time consuming and error prone work. The evaluation of the proposed approach substantiates in the questions

- whether each case can be automated at all and
- whether each case can be defined formally by a process engineer, so that other organizations can make their own modifications of the V-Modell XT or construct even a completely new process model without losing the incremental planning feature.

In the following, each of those cases will be covered by a section showing how a process engineer could express those planning problems with the proposed language – so that the incremental update mechanism (as shown in section 3.2) can do the automation. To spare space, we will occasionally refer to the examples of section 3.1. Please note that each case is more like a class or a type of a problems rather than just a concrete problem. For clarification, consider case 1: the general problem is to schedule (recursively) new artifacts. A concrete problem of this class it the example of scheduling new specification documents. Another example would be to schedule new test documents.

Case 1: Scheduling new Artifacts. At some point in time in a project, an artifact is revised. For instance, a new *system element* is identified and enlisted in the *system architecture* document. Now, following consequent artifacts have to be scheduled:

- A *software specification* document for concretizing the new system element's function, along with a *review* document to check that specification
- An artifact for the new *system element* itself (which would be code and/or binaries)
- A *test specification* and a *test protocol* document, as well as test procedures later on – for each test case identified in the test specification.

With the proposed language at hands, a process engineer can encode these points as shown in fig. 12. Please not that the rules defined in that example do not contain any update instructions as this is a part of the update mechanism.

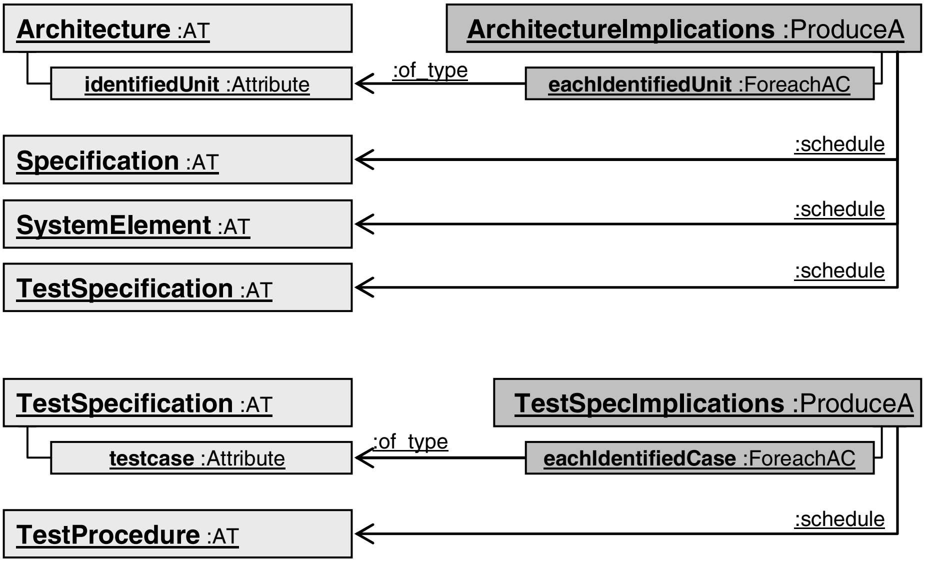


Fig. 12. Rules for encoding the example of case 1

Case 2: Dismissing obsolete Artifacts. As a counterpart to the previous problem, changes may render existing artifacts obsolete. Although removing seems to be a simple operation, it still requires a project manager to look into all document contents and track all dependencies between them. Especially when the project manager is not the only one person, being authorized to make changes. With the proposed language, however, the task of dismissing artifacts has not to be encoded explicitly. This is implicitly given for free by the update mechanism based on triple graph grammars. In short, those rules defined in fig. 12 already contain all necessary information.

Case 3: Check whether everything is up to date. Consider that an interface specification document *d* has been revised by adding an optional parameter *p* to some function therein. Now, there is the need for scheduling a new version for each existing test case, a new version for all interfaces' implementations, along with a corresponding new test protocol version. Right after introducing that new parameter *p* to document *d*, that document can't be rated as finished anymore – at least not until all the consequent versions of dependent artifacts have been produced. This requires a project manager to continuely check all of those artifacts, before he can eventually set document *d* as finished. Thus, calculating and keeping track of artifact states creates an additional overhead. Fig. 9 has already shown how these state changes can be encoded with the language proposed.

Case 4: Planning with Unfinished Artifacts. Taking a second look at artifact states, we see that being "finished" is not the only one state, where consequent actions have to be started. In fact, quality assurance needs some kind of a "submitted" state. Thus, a project manager is not only faced with finished or out-of-date results, but with intermediate ones, too. Until a document currently being revised, is accepted as finished, a project manager has to use the present finished version. This in turn, complicates the way he retrieves artifacts from the artifact library. Fig. 13 shows how rule application can easily be restricted to a specific document state.

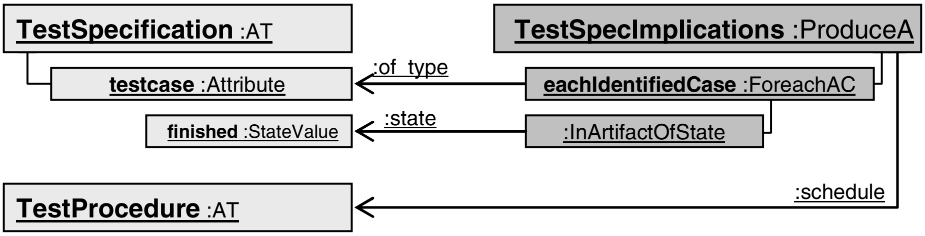


Fig. 13. Example showing how to deal with unfinished artifacts

In addition, some documents might relate their contents to different iterations (or releases if you prefer). So a document rated as "finished" might only imply, that the contents of the current iteration have been finished – without saying a word about the overall project plan. This issue can be expressed by using special document contents for denoting finished and unfinished document parts, as shown in fig. 14.

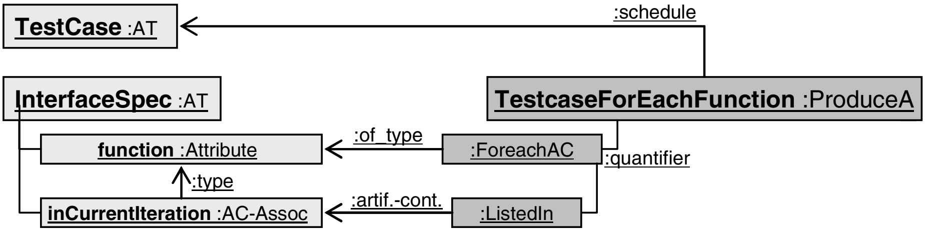


Fig. 14. Example showing how to deal with unfinished artifact parts

Case 5: Dealing with (Adaptive) Process Models. Some artifact types a process model defines are irrelevant for some projects, which imply that no documents need to be produced of that type at all. E.g., a contract is not necessary in an in-house-development. Modern process models, like V-Modell XT [22] or Hermes [32] provide a mechanism called "tailoring", which allows a project manager to adapt a process model to his current project. This way, he can start a project with a minimal set of artifact types he has to take care of. Changing requirements in that project, however, might demand him to readjust his tailoring. This typically results in new artifact types added, along with various corresponding dependencies. Eventually, the project manager has to go through the whole artifact library to find out which of already existing artifacts have to be revised in order to fit the newly tailored process model. Fig. 7 has shown an example of how these dynamics can be encoded.

Another interesting property of the proposed language is that the available constructs can also be used to express static tailoring, too. Initial artifact types can be defined through rules without having any quantifiers as shown in fig. 15. Static tailoring based upon can go through a special initial document named project manual. Project characteristics can be expressed as inputs therein so that tailoring rules can trigger in dependence of possible value assignments.

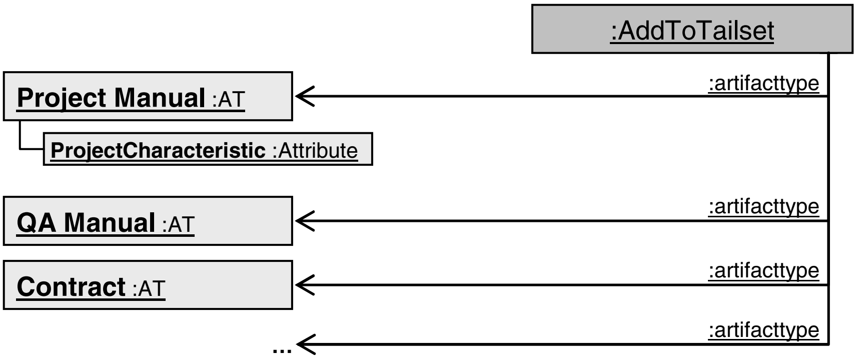


Fig. 15. Example showing how to deal with unfinished artifact parts

Runtime Complexity. By using the big-O calculus (see e.g. [33]), the assessment of a prototype implementation of the proposed approach has shown a polynomial time complexity in relation to project size (counting artifacts and contents), making it practically scalable in general. That polynomial's degree is equal to the maximum arity (measured in the amount of quantifiers) a rule has. However, it can be reduced by incorporating further techniques known from other domains (e.g. partitioning [34]). That is, we organize quantifiable elements into a set of trees derived from their relations and iterate only over resulting tree roots.

5 Conclusions

In this paper, we have presented a process modeling language which can be executed to generate project plans and to update them incrementally – which was the initial motivation. So far, however, state of the art had only one of those two features to offer at once: incremental model transformations allowed updating a generated result without recreating it (and thus without dismissing information added meanwhile). However, using that approach required encoding high-level problems into a low-level language of simple graphs and graph replacement rules. On the other hand, existing process modeling languages (including workflows and change management systems) provided high-level constructs but failed to deliver update mechanisms which can be used right away. Instead, a process engineer had to define his update strategy by himself. In contrast, the proposed solution in this paper provides high-level language constructs which are incrementally updatable, too. This way, a process engineer has not to care about writing update mechanisms himself.

References

1. Tichy, W.F.: RCS—A System for Version Control. Software—Practice and Experience 15(7), 637–654 (1985)
2. Kofmann, M., Perjous, E.: MetaDiff a Model Comparision Framework
3. Lin, Y., Zhang, J., Gray, J.: Model Comparison: A Key Challenge for Transformation Testing and Version Control in Model Driven Software Development. In: Best practices for model-driven software development (2003)
4. Kelter, U., Wehren, J., Niere, J.: A Generic Difference Algorithm for UML Models“ in Software Engineering. LNI, vol. 64, pp. 105–116. GI (2005)
5. Schmidt, M., Gloetzner, T.: Constructing difference tools for models using the SiDiff framework. In: Proceedings of ICSE 2008, pp. 947–948. ACM, Leipzig (2008)
6. Bartelt, C.: An Optimistic Three-way Merge - Based on a Meta-Model Independent Modularization of Models to Support Concurrent Evolution. In: Proceedings of the International Workshop on MoDSE 2008 at CSMR 2008 (2008)
7. Alanen, M., Porres, I.: Difference and Union of Models. In: Stevens, P., Whittle, J., Booch, G. (eds.) UML 2003. LNCS, vol. 2863, pp. 2–17. Springer, Heidelberg (2003)
8. Giese, H., Wagner, R.: Incremental Model Synchronization with Triple Graph Grammars.
9. Schürr, A.: Specification of Graph Translators with Triple Graph Grammars. In: Mayr, S. (ed.) WG 1994. LNCS, vol. 903, pp. 151–163. Springer, Heidelberg (1995)
10. Dang, D., Gogolla, M.: On Integrating OCL and Triple Graph Grammars. In: Models in Software Engineering: Workshops and Symposia At MODELS 2008 (2008)
11. Goldschmidt, T., Uhl, A.: Retainment Rules for Model Transformations”. In: 1st International Workshop on Model Co-Evolution and Consistency Management (2008)
12. Fluri, B.: Assessing Changeability by Investigating the Propagation of Change Types. In: 29th International Conference on Software Engineering, ICSE 2007 (2007)
13. Buckley, C.D., Pulsipher, D.W., Scott, K.: Implementing IBM(R) Rational(R) ClearQuest(R): An End-to-End Deployment Guide. IBM Press (2006)
14. Ma, J., et al.: Customizing Lotus Notes to Build Software Engineering Tools. In: Conference of the Centre for Advanced Studies on Collaborative Research (2003)

15. International Business Machines Corp., ”Jazz” (2009), http://www-306.ibm.com/software/rational/jazz/
16. Osterweil, L.J., et al.: Experience in Using a Process Language to Define Scientific Workflow and Generate Dataset Provenance. In: ACM SIGSOFT 16th International Symposium on Foundations of Software Engineering, November 2008, pp. 319–329 (2008)
17. Sutton Jr., S.M., Osterweil, L.J.: The Design of a Next-Generation Process Language. In: In Proc. of 5th ACM SIGSOFT FSE 5, September 1997, pp. 142–158 (1997) (UM-CS-1997-054)
18. Ternité, T.: Process lines: a product line approach designed for process model development. In: Proceedings of the 35th EUROMICRO SEAA, SPPI Track (2009)
19. Ternité, T., Kuhrmann, M.: Das V-Modell XT 1.3 Metamodell, Technical Report, number TUM-I0905, Technische Universität München (2009)
20. Kuhrmann, M., Ternité, T.: V-Modell XT 1.3 - Neuerungen für Anwender und Prozessingenieure. In: Proceedings of 16 Workshop der Fachgruppe WI-VM der Gesellschaft für Informatik e.V., April 2009, Germany Shaker Verlag (2009)
21. Object Management Group (OMG), Software Process Engineering Meta-Model, version 2.0, (April 2008)
22. KBSt, Official Website of the V-Model XT (2009), http://www.v-modell-xt.de
23. W. Kranz, D. Rauh, flyXT – Das neue Vorgehensmodell der EADS DE“, In Proceedings of Software & Systems Engineering Essentials, Berlin, 2009.
24. Conradi, R., et al.: EPOS: Object-oriented cooperative process modelling. In: Software Process Modelling and Technology, pp. 33–70. John Wiley & Sons Inc, Chichester (1994)
25. Fernström, C.: PROCESS WEAVER: Adding process support to UNIX. In: Proceedings of ICSP, pp. 12–26 (1993)
26. Junkermann, G., et al.: MERLIN: Supporting cooperation in software development through a knowledge-based environment. In: Software Process Modelling and Technology, pp. 103–129. John Wiley & Sons Inc., Chichester (1994)
27. Deiters, W., Gruhn, V.: Managing software processes in the environment melmac. In: Proc. Of the Fourth ACM SIGSOFT Symposium on Practical Software Development Environments (1990)
28. Kaiser, G.E., et al.: Experience with process modeling in the MARVEL software development environment kernel. In: Proceedings of 23rd HICCS (1990)
29. Workflow Management Coalition. Workflow Standard, Workflow Process Definition Interface – XML Process Definition Language, Document Number WFMCTC-1025 (2002)
30. Kindler, E., Wagner, R.: Triple Graph Grammars: Concepts, Extensions, Implementations, and Application Scenarios, Technical report, University of Paderborn (2007)
31. Giese, H., Hildebrandt, S.: Incremental model synchronization for multiple updates. In: Proceedings of GRaMoT 2008. ACM, New York (2008)
32. Federal Administration of Switzerland, The HERMES Method (2007), http://www.hermes.admin.ch
33. Sipser, M.: Introduction to the Theory of Computation. PWS Publishing. In: Measuring complexity section 7.1, pp. 226–228. PWS Publishing (1997) ISBN 0-534-94728-X
34. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE Transactions on Image Processing, 561–576 (2000)

A Software Metric for Python Language

Sanjay Misra and Ferid Cafer

Department of Computer Engineering, School of Information and Communication Technology,
Federal University of Technology, Minna, Nigeria
Department of Software Engineering, Atilim University, Ankara, Turkey
ssopam@gmail.com, fcafer@atilim.edu.tr

Abstract. There are many metrics for evaluating the quality of codes written in different programming languages. However, no efforts have been done to propose metrics for Python, which is an important and useful language especially for the software development for the embedded systems. In this present work, we are trying to investigate all the factors, which are responsible for increasing the complexity of code written in Python language. Accordingly, we have proposed a unified metric for this language. Practical applicability of the metric is demonstrated on a case study.

Keywords: Software complexity, complexity metrics, Python, software development.

1 Introduction

Fenton defines measurement as ‘Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined unambiguous rules’[6]. In software engineering software metrics are the only tools to control the quality of software. Furthermore, requirement to improve the software quality is the prime objective, which promoted the research projects on software metrics technology. It is always hard to control the quality if the code is complex. Complex codes always create problems to software community. It is hard to review, test, maintain as well as managing such codes. It is difficult to manage also. As a consequence, those handicaps increase the maintenance cost and the cost of the product. Due to these reasons, it is strongly recommened to control the complexity of the code from the beginning of the software development process. Software metrics help to achieve this goal.

In the past, researchers proposed their methodologies for evaluating codes, which were written in procedural languages [12], such as C. Later, studies focused on object-oriented (OO) programming languages, e.g. C++ [1, 2, and 3] and Java. Software metrics for other languages and technologies such as XML and Web services [13, 14, and 24] can also be found in the literature. However, to evaluate codes written in Python language has not found proper attention as we were expecting. In fact, today, Python is not as popular as Java and C++ but due to its unique features, it is a more comfortable language for software development. It is gaining popularity day by day in

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 301–313, 2010.

the software community. Several conferences and workshops devoted to Python including SciPy India 2009[16], RuPy '09[17], Poland, FRUncon 09[18], USA, ConFoo.ca 2010[19], Canada, are proving the importance of Python. It is not the end of the success stories of Python. It is highly understandable in comparsion to other OO languages, hence less expensive to maintain. Famous enterprises like Google and YouTube choose use Python.

One way to evaluate the complexity of the Python code is through the metrics developed for procedural languages. However, all the available metrics cover only special features of the language. For example, if we apply line of code then only size will be considered, if we apply McCabe's Complexity metric, we will only cover the control flow of the program. In addition, the metrics applicable to the procedural languages do not fit to the modern languages such as C++ [3]. By keeping all these issue in mind, in the present work, we are evaluating the complexity of Python code by identifying all the factors, which may be play important roles in the complexity of the code. Although we have tried to include most of those factors, it is possible to add more. In fact, Python is an OO language hence it includes most of the features of other OO languages; however, differences occur in the main program body. In addition to that, execution and dynamic typing provide big productivity gains over Java; Python programs need less extraneous endeavour (i.e. cleaner code) [21].

The paper is organized in the following way. We discuss the importance of Python language and the available metrics in Section 2. Following that, we propose our metric in Section 3. The metric is demonstrated and compared with other metrics in Section 4. The metric is validated theoretically in Section 4. The conclusion drawn on this work is in Section 5.

2 The Literature Survey

Python is a programming language that lets the programmer to work more quickly to integrate systems more effectively [10, 11]. It is a free of charge language for commercial purpose. It runs on all major operating systems including Windows, Linux/Unix, Mac OS X, and also has been ported to the Java and .NET virtual machines [10]. Besides to those features, Python is an effective language especially for the software development for embedded systems. It can also be used in web development [4, 5]. It is an ideal language to solve problems, especially on Linux and UNIX, for building software applications in life science research and development, and processing in natural languages [7-9].

In case of embedded system, where inexpensive components and maintenance are demanded, Python may provide best solution. With Python, one can achieve these goals in terms of small size, high reliability and low power consumption. In addition, the developers who have background of Java, C and/or Visual Basic [4] can learn it without major effort. In fact, it offers features of mixture of programming languages. It provides most of the features of OO languages in a powerful and simple way.

In addition to these powerful features of this language, no proper techniques are available to evaluate the quality of Python code to our knowledge. We could not find a single metric in a published form except some online articles [25-30]. These available articles are related to the available tools, which are limited to calculate the simple metrics. For example, Pythius [26] tool calculates the complexity of Python Code by computing simple metrics such as ratio of comments to code lines, module and function size.

Another tool ‘snakefood’ proposed by Martin Blais [27] provides the dependency graphs for Python. It can be useful to calculate McCabe’s Cyclomatic complexity for Python code. Pygenie tool developed by David Stanek [28] also calculates the McCabe’s Cyclomatic complexity for Python code. Reg Charney has also developed an open source code complexity measurement tool named as PyMetrics [29] which is capable of counting the following metrics: block count, maximum block depth, number of doc strings for Python classes, number of classes, number of comments, number of inline comments, total number of doc strings, number of function doc strings, number of functions, number of keywords used, number of lines, number of characters and number of multiple exit functions. Another tool available online is Pyntch [30]. Pinch *(*PYthoN Type CHecker) is a static code analyzer which detects possible run time errors before actually running a code.

All of the above tools are effective in evaluating the quality of the Python language only up to an extent. Most of them are confined to compute simple metrics, which give only idea for some specific attributes; none of them are capable to evalute majority of attributes in a single metric.

3 Proposed Metric

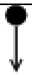
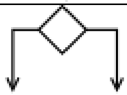
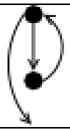
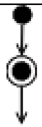
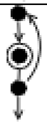
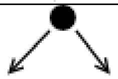
Based on the drawbacks as discussed in the previous section we are proposing a new metric that is intended to measure the complexity of the Python code, by including most of the parameters, which are responsible for the overall complexity. The components/factors of our metrics are:

1. Variables or attributes,
2. Basic control structures; and
3. Classes

Variables and attributes: They are one of the causes of complexity. Further, if a variable’s name is arbitrarily given, then the comprehensibility of that code will be lower [20]. Therefore, variables and attributes of classes should have meaningful names. Although, it is suggested that the name of the variables should be chosen in such a way that it is meaningful in programming, there are developers who do not follow this advice strictly. If the variable names are taken arbitrarily, it may not be a grave problem if the developer himself evaluates the code. However, it is not the case in the real life. After the system is developed, especially during maintenance time, arbitrarily named variables may increase the difficulty in understanding four times more [20] than the meaningful names.

Basic Control Structures (BCS’s): Sequences’ complexity depends on its mathematical expression [15]. Functions help tidiness of a code. Also, they may increase the reusability of the code. However, each function call disturbs the fluency of reading a code. Loops are used to repeat a statement for more than one time, although, they decrease computing performance. Especially, the more nested loops there are, the more time takes to run the code. Also, human brain has similarities with computers in interpreting a data [22], [23]. Conditional statements are used to make a program dynamic. On the other hand, by presenting more combinations they decrease the easiness of grasping the integrity of a program. Because, conditional statements can be thought to be sequences build up in different possible situations. If the conditional situations are nested, then the complexity much becomes higher. The situation in exceptions is similar. We are assigning the weights for each basic control structure followed by the similar approach of Wang [15]. Wang proved and assigned the weights for sub conscious function, meta cognitive function and higher cognitive function as 1, 2 and 3 respectively. Although, we followed the similar approach with Wang, we made some modifications in the weights of some Basic Control Structures as shown in Table 1.

Table 1. Values of Structures

Category	Value	Flow Graph
Sequence	1	
Condition	2	
Loop	3	
Nested Loop	3^n	-
Function	2	
Recursion	3	
Exception	2	

Classes: The complexity of a class is proposed by considering the complexity due to attributes and methods (functions). Furthermore, for calculating the complexity due to attributes and methods, we follow the guidelines given in sections 3.1 and 3.2.

The Metric

We develop our metric depending upon the structural values given in Table 1.

Software Metric for Python (SMPy)
= CIclass+CDclass+Cglobal+Ccoupling (1)

Where, CIclass = Complexity due to Inheritance
CDclass = Complexity of Distinct Class
Cglobal= Global Complexity
Ccoupling= Complexity due to coupling between classes**.**

All these factors are defined as follows:

Cclass can be defined as;
Cclass = weight(attributes)+ weight(variables)+ weight(structures)+ weight(objects)- weight(cohesion) (2)
Where, Cclass = Complexity of Class

Where, weight of attributes W (attribute) is defined as:

W (attributes) = 4*AND+MND. (2.1)

Where, AND = Number of Arbitrarily Named Distinct Variables/Attributes
MND = Number of Meaningfully Named Distinct Variables/Attributes
Weight of variable W(variables) is defined as:

W(variables) = 4*AND+MND (2.2)

Weight of structure W(structures) is defined as:

W(structures) = W(BCS) (2.3)

Where, BCS are basic control structure.
Weight of objects W(object) is defined as:

W(objects) = 2 (2.4)

Creating an object is counted as 2, because while creating a function constructor is automatically called. Therefore, it is the same as calling a function or creating an object. Here it is meant to be the objects created inside a class.

Weight of cohesion is defined as:
W(cohesion) = MA/AM
Where, MA = Number of methods where attributes are used
AM = Number of attributes used inside methods

While counting the number of attributes there is no any importance of AND or MND.

Notes:
Function call: during inheritance, calling super class's constructor is not counted.
Global variable: static attribute is counted as a global variable.

Cglobal can be defined as;

Cglobal=W(variables+structures+objects) (3)

weight of variable W(variable) is defined as:

$$\text{W(variables)} = 4*\text{AND}+\text{MND} \tag{3.1}$$

The variables are defined globally.

Weight of structure W(structure) is defined as:

$$\text{W(structures)} = \text{W(BCS)}+\text{obj.method} \tag{3.2}$$

Where, BCS are basic control structure, and those structures are used globally. ‘obj.method’ calls a reachable method of a class using an object. ‘obj.method’ is counted as 2, because it calls a function written by the programmer.

weight of objects W(object) is defined as:

$$\text{W(objects)} = 2 \tag{3.3}$$

Creating an object is counted as 2, as it is described above. Here it is meant to be the objects created globally or inside any function which is not a part of any class.

Notes:

Exception: while calculating try catch statement, only the number of “catch”es are counted as 2. “try” itself is not counted.

CIclass can be defined as;

There are two cases for calculating the complexity of the Inheritance classes depending on the architecture:

- If the classes are in the same level then their weights are added.
- If they are children of a class then their weights are multiplied due to inheritance property.

If there are m levels of depth in the OO code and level j has n classes then the complexity of the system due to inheritance is given as;

$$\text{CIclasses} = \prod_{j=1}^{m} \left[\sum_{k=1}^{n} CC_{jk} \right] \tag{4}$$

CDclass can be defined as;

$$\text{CDclass}=\text{Cclass(x)} + \text{Class(y)} +\ldots \tag{5}$$

Note: all classes, which are neither inherited nor derived from another, are part of CDclass even if they have caused coupling together with other classes.

Coupling is defined as;

$$\text{Coupling} = 2^{c} \tag{6}$$

c = Number of Connections

A method which calls another method in another class creates coupling. However, if a method of a class calls the method of its super class, then it is not considered to be coupling. In order to provide a connection there has to be two entities. It means one connection is in between two points. Thus the base number is taken as 2. Another reason for that is function call has a value of 2; c is total number of connection made from one method to other method(s) in another class (es). It is taken as 'to the power', because each connection makes a significant increment in cognitive comprehensibility in software.

4 Demonstration of the Metric

We have demonstrated our proposed complexity metric given by equation 1 by a programming example written in Python language. The complete code for the following figure is given in the Appendix.
Example Class Diagram:

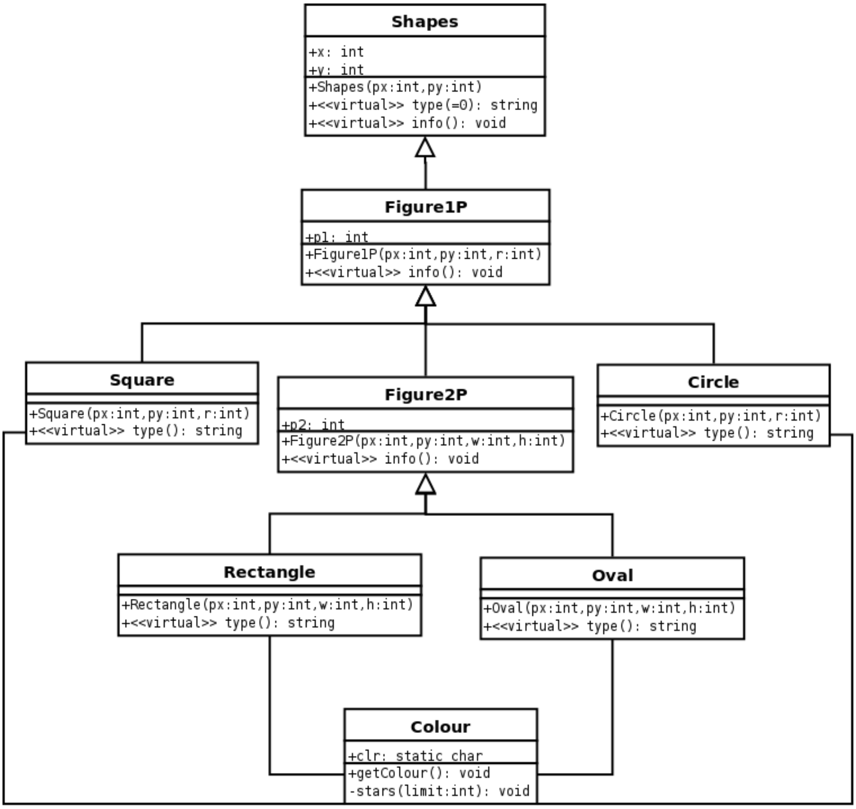

Shapes
+x: int
+y: int
+Shapes(px:int,py:int)
+<<virtual>> type(=0): string
+<<virtual>> info(): void
Figure1P
+p1: int
+Figure1P(px:int,py:int,r:int)
+<<virtual>> info(): void
Square
+Square(px:int,py:int,r:int)
+<<virtual>> type(): string
Figure2P
+p2: int
+Figure2P(px:int,py:int,w:int,h:int)
+<<virtual>> info(): void
Circle
+Circle(px:int,py:int,r:int)
+<<virtual>> type(): string
Rectangle
+Rectangle(px:int,py:int,w:int,h:int)
+<<virtual>> type(): string
Oval
+Oval(px:int,py:int,w:int,h:int)
+<<virtual>> type(): string
Colour
+clr: static char
+getColour(): void
-stars(limit:int): void

Fig. 1. Shapes – Class Diagram

The proposed example has eight classes. The components of our metric for all classes are summarized in Table 1. Based on these values we have calculated Cclass, CIclass, CDclass, Cglobal, coupling values of the system. It is worth to mention that during the calculation of complexity of inheritance, we should be careful in computing of CIclass. We have to add the complexity of the classes at the same level and only multiply with their parent classes, as shown in the following.

Table 2. Class Complexity

class	attribute	string	variable	object	MA	AM	cohesion	Cclass
Colour	0	33	2	0	0	0	0	35
Shapes	2	6	0	0	2	2	1	7
Figure1P	1	8	0	0	2	1	2	7
Square	0	27	0	2	0	0	0	29
Circle	0	27	0	2	0	0	0	29
Figure2P	1	11	0	0	1	1	1	10
Rectangle	0	27	0	2	0	0	0	29
Oval	0	27	0	2	0	0	0	29

Cclass(Colour)=35
Cclass(Shapes)=7
Cclass(Figure1P)=7
Cclass(Square)=29
Cclass(Circle)=29
Cclass(Figure2P)=10
Cclass(Rectangle)=29
Cclass(Oval)=29

Table 3. Non-Class Complexity

Non-Class	var+str+obj	Complexity
Cglobal	24	24

CIclass=Shapes*(Figure1P*(Square+Circle+Figure2P*(Rectangle+Oval)))
=7*(7*(29+29+10*(29+29)))
=31262
CDclass=35
Cglobal=24

SMPy=CIclass+CDclass+Cglobal+coupling
$=31262+35+24+2^4$
=31337

Calculation is as follows;

1. Complexity of each class was calculated. Attributes, methods, variables, objects, structures, and cohesion were included.
2. Complexity of global structure was calculated. Variables, objects, structures, functions, and the main function were included.
3. Classes were separated as inside inheritance and distinct.

4. Complexity of inheritance was calculated. Super class was multiplied by the summation of the classes which are derived from it.
5. Complexity of inheritance, complexity of distinct class, complexity of global structure, and coupling were summed to reach the result of SMPy.

5 Conclusion

This paper presents a complexity metric for Python language. It is observed that there is lack of proper metrics for this useful and important language. This factor motivates us to work on it and produce such a metric, which can cover all the aspects of complexity. Consequently, a new approach, which is unification of all the attributes, is presented. Furthermore, we think the proposed metric can be applied to any OO language. It can be applicable to the procedural language by omitting the terms responsible for OO features. We hope that the present work will find attention from the researchers and practitioners, who are working in OO domain, especially who are using Python.

References

1. Costagliola, G., Tortora, G.: Class points: An approach for the size Estimation of Object-oriented systems. IEEE Transactions on Software Engineering 31(1), 52–74 (2005)
2. Misra, S., Akman, I.: Weighted Class Complexity: A Measure of Complexity for Object Oriented Systems. Journal of Information Science and Engineering 24, 1689–1708 (2008)
3. Chidamber, S.R., Kermer, C.F.: A Metric Suite for object oriented design. IEEE Transacations Software Engineering SE-6, 476–493 (1994)
4. http://www.Python.org/about/success/carmanah/
5. Lutz, M.: Learning Python, 4th edn., Ebook, Safari Books Online, O'Reilly Media, Sebastopol (2009)
6. Fenton, N.E., Pfleeger, S.L.: Software Metrics: A Rigorous and Practical Approach, 2nd Revised edn. PWS Publishing, Boston (1997)
7. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python, 1st edn., Ebook, Safari Books Online, O'Reilly Media, Sebastopol (2009)
8. Gift, N., Jones, J.M.: Python for Unix and Linux System Administration, 1st edn., Ebook, Safari Books Online, O'Reilly Media, Sebastopol (August 2008)
9. Model, M.L., Tisdall, J.: Bioinformatics Programming Using Python, 1st ed., Ebook, Safari Books Online. O'Reilly Media, Sebastopol (2009)
10. Python Programming Language, http://www.Python.org/ cited 04.10.2009
11. Lutz, M.: Programming Python, 3rd ed., Ebook, Safari Books Online. O'Reilly Media (2006)
12. Misra, S., Akman, I.: Unified Complexity Metric: A measure of Complexity. In: Proc. of National Academy of Sciences Section A, 2010 (in press)
13. Basci, D., Misra, S.: Data Complexity Metrics for Web-Services. Advances in Electrical and Computer Engineering 9(2), 9–15 (2009)
14. Basci, D., Misra, S.: Measuring and Evaluating a Design Complexity Metric for XML Schema Documents. Journal of Information Science and Engineering, 1415–1425 (September 2009)

15. Wang, Y., Shao, J.: A New Measure of Software Complexity Based on Cognitive Weights. Can. J. Elec. Comput. Engg., 69–74 (2003)
16. SciPy.in 2009, http://scipy.in/ cited 16.10.2009
17. Rupy 2009. http://rupy.eu/ cited 10.11.2009
18. FrontRangePythoneersUc 2009,
    http://wiki.python.org/moin/FrontRangePythoneersUc09
    cited 05.10.2009
19. Confoo.Ca Web Techno Conference, http://www.confoo.ca/en cited 14.11.2009
20. Kushwaha, D.S., Misra, A.K.: Improved Cognitive Information Complexity Measure: A Metric That Establishes Program Comprehension Effort. SIGSOFT Software Engineering Notes 31(5) 1–7 (2006)
21. DMH2000 C/Java/Python/Ruby, http://www.dmh2000.com/cjpr/, cited 15.10.2009
22. Neuroscience – Brain vs. Computer,
    http://faculty.washington.edu/chudler/bvc.html, cited 17.10.2009.
23. Computer vs. The Brain,
    http://library.thinkquest.org/C001501/the_saga/sim.htm,
    cited 17.10.2009
24. Basci, D., Misra, S.: Entropy metric for XML DTD documents. SIGSOFT Softw. Eng. Notes 33(4), 1–6 (2008)
25. Python Code Complexity Metrics and Tools available from: http://agiletesting.blogspot.com/2008/03/Python-code-complexity-metrics-and.html
26. Pythius Homepage, http://pythius.sourceforge.net/, cited 03.10.2009)
27. Python Dependency Graphs, http://furius.ca/snakefood/, cited 08.11.2009
28. Measuring Cyclomatic Complexity of Python Code
    http://www.traceback.org/2008/03/31/measuring-cyclomatic-complexity-of-Python-code/
29. PyMetrics, http://sourceforge.net/projects/pymetrics/, cited 21.09.2009
30. Shinyama, Y.: http://www.unixuser.org/~euske/python/index.html, cited 06.10.2009

Appendix

```
//Shapes program written in C++ to illustrate the usability of the proposed metric
   #include <iostream>
   #include <string>
   using namespace std;
   // Colour
   class Colour{
          void stars(int limit);
   public:
          static char c;
          void getColour();
   };
   void Colour::getColour(){
          if (c=='s')
                   cout<<"Yellow"<<endl;
          else if (c=='c')
```

```
                cout<<"Violet"<<endl;
        else if (c=='r')
                cout<<"Red"<<endl;
        else if (c=='o')
                cout<<"Orange"<<endl;
        else
                cout<<"White"<<endl;
        stars(5);
}
void Colour::stars(int limit){
        int outer_loop, inner_loop;
        for (outer_loop=limit; outer_loop>0; outer_loop--){
                for (inner_loop=1; inner_loop<=outer_loop; inner_loop++)
                        printf(“*”);
                printf(“\n”);
        }
}
// -----------
char Colour::c;

class Shapes {
public:
        Shapes(int px, int py):x(px),y(py) {}
        int x, y; //position
        virtual string type() = 0;
        virtual void info() {
                cout << endl << "figure: " << type() << endl;
                cout << "position: x=" << x << ", y=" << y << endl;
        }
};

class Figure1P : public Shapes {
public:
        Figure1P(int px, int py, int r):p1(r),Shapes(px, py) {}
        int p1;
        virtual void info() {
                Shapes::info();
                cout << "property 1: p=" << p1 << endl;
        }
};

class Square : public Figure1P {
public:
        Colour *its_colour;
        Square(int px, int py, int r):Figure1P(px, py, r) {}
        virtual string type() {
                Colour::c='s';
                its_colour->getColour();
                return "square";
        }
};

class Circle : public Figure1P {
public:
        Colour *its_colour;
        Circle(int px, int py, int r):Figure1P(px, py, r) {}
        virtual string type() {
                Colour::c='c';
                its_colour->getColour();
                return "circle";
        }
};

class Figure2P : public Figure1P {
public:
        Figure2P(int px, int py, int w, int h):p2(h),Figure1P(px, py, w) {}
        int p2;
        virtual void info() {
                Figure1P::info();
                cout << "property 2: p=" << p2 << endl;
        }
};

class Rectangle : public Figure2P {
public:
        Colour *its_colour;
        Rectangle(int px, int py, int w, int h):Figure2P(px, py, w, h) {}
        virtual string type() {
                Colour::c='r';
                its_colour->getColour();
                return "rectangle";
        }
};

class Oval : public Figure2P {
public:
        Colour *its_colour;
        Oval(int px, int py, int w, int h):Figure2P(px, py, w, h) {}
        virtual string type() {

                Colour::c='o';
                its_colour->getColour();
                return "oval";
        }
```

```
    };

    // Freeing memory
    void freeRAM(Shapes *objs[], int i){

            delete objs[i];

    }
    // --------------

    int main(void) {
            Shapes **objs = new Shapes*[5];
            // creating objects
            objs[0] = new Circle(7, 6, 55);
            objs[1] = new Rectangle(12, 54, 21, 14);
            objs[2] = new Square(19, 32, 10);
            objs[3] = new Oval(43, 10, 4, 3);
            objs[4] = new Square(3, 41, 3);
            bool flag=false;
            do {
                    cout << endl << "We have 5 objects with numbers 0..4" << endl;
                    cout << "Enter object number to view information about it " <<
endl;
                    cout << "Enter any other number to quit " << endl;
                    char onum; // in fact, this is a character, not a number
                    // this allows user to enter letter and quit...
                    cin >> onum;
                    // flag -- user have entered number 0..4
                    flag = ((onum >= '0')&&(onum <= '4'));
                    if (flag)
                            objs[onum-'0']->info();
            } while(flag);

            for(int i=0;i<5;i++)
                    freeRAM(objs,i);
            delete [] objs;
            return(0);
      }
```

Certifying Algorithms for the Path Cover and Related Problems on Interval Graphs

Ruo-Wei Hung[1],* and Maw-Shang Chang[2]

[1] Department of Computer Science and Information Engineering,
Chaoyang University of Technology,
Wufong, Taichung 413, Taiwan
rwhung@cyut.edu.tw

[2] Department of Computer Science and Information Engineering,
National Chung Cheng University,
Ming-Hsiung, Chiayi 621, Taiwan
mschang@cs.ccu.edu.tw

Abstract. A certifying algorithm for a problem is an algorithm that provides a certificate with each answer that it produces. The certificate is a piece of evidence that proves the answer has no compromised by a bug in the implementation. A Hamiltonian cycle in a graph is a simple cycle in which each vertex of the graph appears exactly once. The Hamiltonian cycle problem is to test whether a graph has a Hamiltonian cycle. A path cover of a graph is a family of vertex-disjoint paths that covers all vertices of the graph. The path cover problem is to find a path cover of a graph with minimum cardinality. The scattering number of a noncomplete connected graph $G = (V, E)$ is defined by $s(G) = \max\{\omega(G-S)-|S| : S \subseteq V$ and $\omega(G-S) \geqslant 1\}$, in which $\omega(G-S)$ denotes the number of components of the graph $G-S$. The scattering number problem is to determine the scattering number of a graph. A recognition problem of graphs is to decide whether a given input graph has a certain property. To the best of our knowledge, most published certifying algorithms are to solve the recognition problems for special classes of graphs. This paper presents $O(n)$-time certifying algorithms for the above three problems, including Hamiltonian cycle problem, path cover problem, and scattering number problem, on interval graphs given a set of n intervals with endpoints sorted. The certificates provided by our algorithms can be authenticated in $O(n)$ time.

Keywords: Certifying algorithm, path cover, Hamiltonian cycle, scattering number, interval graph.

1 Introduction

The study of certifying algorithms is motivated by software engineering, software reliability and the insight that software is often not bug-free. Although an

* Corresponding author.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 314–323, 2010.

algorithm has been always proved to be correct, its implementation may contain bugs. Thus, it is desirable to have tools for knowing whether the output of an implementation of an algorithm is correct or returned due to a bug. Obviously, there is no way to guarantee by the design and analysis of an algorithm that its implementations are bug-free. Nevertheless, certifying algorithm design may support software reliability.

The name "certifying algorithm" was introduced by Kratsch et al. in [13]. A *certifying algorithm* for a problem is an algorithm that provides a *certificate* with each answer that it produces. An *authentication algorithm* is a separate algorithm that confirms the validity of the answer by checking the certificate; it takes the input, the answer, and the certificate produced by the original algorithm, and verifies (independently of the original algorithm) whether the answer is correct. For example, an implementation of a certifying algorithm testing whether an input graph is bipartite provides an odd cycle as a certificate whenever it claims the input graph is not bipartite, and provides two disjoint independent vertex sets as a certificate if it claims the input graph is bipartite. An authentication algorithm can verify the correctness of an answer claiming the input graph is not bipartite by checking whether the certificate is an odd cycle of the input graph indeed; and verify the correctness of an answer claiming the input graph is bipartite by checking whether the certificate does consist of two disjoint independent sets whose union is the vertex set of the input graph. Usually we prove the correctness of our algorithm after we design it. However the purpose of a certificate produced by a certifying algorithm is not to prove the correctness of the algorithm. It is for implementation of the algorithm. When we implement a valid algorithm, we are not sure that the implementation is bug-free. One way to see the correctness of the implementation is to output a certificate for each answer it produces and let an authentication program verify the correctness of the answer by checking the certificate. Certifying algorithms reduce the risk of erroneous answer, caused by bugs in the implementation. For more background on certifying algorithms, we refer readers to [11,14].

A *recognition algorithm* is an algorithm that decides whether some given input (graph, geometrical object, picture, etc.) has a certain property. Such an algorithm accepts the input if it has the property or rejects it if it does not. There are some certifying recognition algorithms appeared in the literatures recently [6,10,11,14]. In fact some recognition algorithms published before the term of certifying algorithms appeared are certifying algorithms already. To the best of our knowledge, most published certifying algorithms are recognition algorithms for special classes of graphs. In this paper, we give the first certifying algorithm for an optimization problem, i.e., the path cover problem on interval graphs. We also give certifying algorithms for the Hamiltonian cycle and scattering number problems on interval graphs.

All graphs considered in the paper are finite and undirected, without loops or multiple edges. Throughout this paper, let m and n denote the number of edges and the number of vertices of a graph, respectively. A *Hamiltonian cycle* in a graph is a simple cycle in which each vertex of the graph appears exactly once.

A *Hamiltonian path* in a graph is a simple path with the same property. The *Hamiltonian cycle* (resp. *path*) *problem* involves testing whether or not a graph contains a Hamiltonian cycle (resp. path). The *Hamiltonian problems* include the Hamiltonian path and Hamiltonian cycle problems and have numerous applications in different areas, including establishing transport routes, production launching, the on-line optimization of flexible manufacturing systems [2], pattern recognition [19], etc. It is well known that the Hamiltonian problems are NP-complete for general graphs [8]. A *path cover* of a graph G is a family of vertex-disjoint paths that covers all vertices of G. The *path cover problem* is to find a path cover of a graph G with minimum cardinality, denoted by $\pi(G)$. It is evident that the path cover problem for general graphs is NP-complete since finding a path cover, consisting of a single path, corresponds directly to the Hamiltonian path problem. The path cover problem has many practical applications including code optimization [3], mapping parallel programs to parallel architectures [17], program testing [16], etc. Let $G = (V, E)$ be a noncomplete connected graph. The *scattering number* of G was introduced by Jung in [12] and is defined by $s(G) = \max\{\omega(G-S)-|S| : S \subseteq V \text{ and } \omega(G-S) \geqslant 1\}$, where $G-S$ denotes the subgraph of G induced by $V - S$ and $\omega(G - S)$ denotes the number of connected components of $G - S$. A set $S \subseteq V$ with $\omega(G - S) - |S| = s(G)$ and $\omega(G - S) \geqslant 1$ is called a *scattering set* of G. By convention, the scattering number of a complete graph $K = (V, E)$ is $-|V|$ and V is the unique scattering set. Note that if G is not connected then $s(G) \geqslant \omega(G)$. The *scattering number problem* is to determine a scattering number of a graph. The scattering number of a graph is strongly related to hamiltonian properties of graphs. If G has a Hamiltonian cycle, then $s(G) \leqslant 0$; and if G has a Hamiltonian path, then $s(G) \leqslant 1$ [4,7]. On the other hand, $\pi(G) \geqslant \max\{1, s(G)\}$ holds for arbitrary graph G by the pigeonhole principle, since each connected subgraph needs to be covered by at least one path.

An interval family I is a collection of intervals on a real line. A graph G is an *interval graph* if there exist an interval family I and a one-to-one mapping of the vertices of G and the intervals in I such that two vertices in G are adjacent if and only if their corresponding intervals in I intersect. We call I an *interval model* of G. For an interval model I, we use $G(I)$ to denote the interval graph for I. Interval graphs have a variety of applications involving VLSI design, scheduling [9], and genetics [20]. Kratsch et al. [14] gave a linear-time certifying recognition algorithm for interval graphs.

Arikati and Rangan [1] presented an $O(n+m)$-time algorithm to solve the path cover problem on interval graphs. Manacher et al. [15] gave an $O(n \log \log n)$-time algorithm for the Hamiltonian cycle problem on a set of n endpoint-sorted intervals. Chang et al. [5] proposed $O(n)$-time algorithms for both the Hamiltonian cycle and path cover problems on interval graphs given a set of n endpoint-sorted intervals. Unfortunately, their algorithms fail to provide supporting evidence for their answers. In this paper, we will propose $O(n)$-time certifying algorithms for the path cover and Hamiltonian cycle problems on interval graphs given a set of n endpoint-sorted intervals. Further, we show that the above two algorithms can be used to design a certifying algorithm for computing the scattering number of a interval graph. The certificates provided by our algorithms can be authenticated in $O(n)$ time.

2 Terminology and Background Results

We start with some notations. For two sets X and Y, let $X - Y$ denote the set of elements of X that are not in Y. Let $G = (V, E)$ be a graph with vertex set V and edge set E. A path P in G, denoted by $v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_{|P|-1} \rightarrow v_{|P|}$, is a sequence $(v_1, v_2, \ldots, v_{|P|-1}, v_{|P|})$ of vertices such that $(v_i, v_{i+1}) \in E$ for $1 \leqslant i < |P|$. The first and last vertices visited by P are called the *path-start* and *path-end* of P, denoted by $start(P)$ and $end(P)$, respectively. We use $v_i \in P$ to denote "P visits v_i". In addition, we will use P to refer to the set of vertices visited by path P if it is understood without ambiguity.

For a graph $G = (V, E)$ and a subset $S \subseteq V$ of vertices, we write $G[S]$ for the subgraph of G *induced* by S, $G - S$ for the subgraph $G[V - S]$, i.e., the subgraph induced by $V - S$. In addition, we write $\omega(G - S)$ for the number of connected components of $G - S$. Let S and C be two disjoint subsets of vertices in G such that S is nonempty. We say that S is an *island with respect to C in G* or an *island in $G - C$* if no vertex in S is adjacent to any vertex of $V - (C \cup S)$ in G. By the above definition, an island S with respect to C in G is not empty and it contains at least one connected component in $G - C$. A subset C of vertices of G is called a *cutset* if the removal of C from G disconnects G.

Since each connected subgraph needs to be covered by at least one path, the following propositions can be easily verified by the pigeonhole principle.

Proposition 1. *[18] Let C be a cutset of a connected graph G and let g be the number of connected components in $G - C$. Then, $\pi(G) \geqslant g - |C|$.*

Proposition 2. *Let C be a cutset of a connected graph G and let g be the number of connected components in $G - C$. If $g > |C|$, then G has no Hamiltonian cycle.*

The following theorem states well-known necessary conditions for hamiltonian properties.

Theorem 1. *[7] If G has a Hamiltonian cycle, then $s(G) \leqslant 0$; and if G has a Hamiltonian path, then $s(G) \leqslant 1$.*

Since each connected subgraph needs to be covered by at least one path, the following proposition immediately holds.

Proposition 3. *For any graph G, $\pi(G) \geqslant \max\{1, s(G)\}$.*

3 A Certifying Algorithm for the Path Cover Problem on Interval Graphs

Arikati and Rangan [1] gave an $O(n+m)$-time algorithm for the path cover problem on interval graphs. Manacher et al. [15] presented an $O(n \log \log n)$-time for

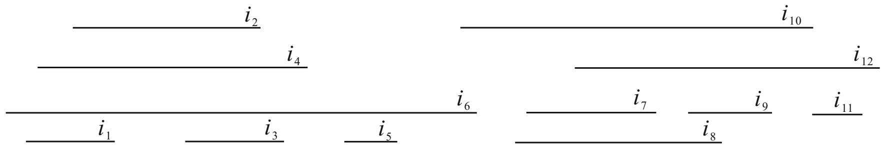


Fig. 1. An interval model I of twelve endpoint-sorted intervals

the Hamiltonian path problem on interval graphs given a set of n endpoint-sorted intervals. The two papers were published in the same volume of *Information Processing Letters*. They used the same greedy approach independently. Assume that the input graph is given by an interval model I that is a set of n endpoint-sorted intervals labeled by $1, 2, \ldots, n$ in increasing order of their right endpoints. Notice that we do not distinguish an interval from its label. The left endpoint of interval x is denoted by $\mathit{left}(x)$ and the right endpoint by $\mathit{right}(x)$. Interval x is denoted by $(\mathit{left}(x), \mathit{right}(x))$. An interval x is said to *contain* another interval y if every point of y falls within the interior of $(\mathit{left}(x), \mathit{right}(x))$. For convenience, we need the following notation.
(1) For two distinct intervals x, y in I, x is smaller than y (or y is larger than x), denoted by $x < y$, if $\mathit{right}(y)$ is to the right of $\mathit{right}(x)$, and y is to the right of x if $\mathit{left}(y)$ is to the right of $\mathit{right}(x)$.
(2) $s(I)$ denotes the interval in I with the leftmost right endpoint; that is, $s(I) \leqslant x$ for $x \in I$.

We first introduce Procedure **GP** that is the key procedure used in the algorithm in [1] for the path cover problem. Given an interval model I, Procedure **GP** uses a greedy principle to obtain a path Z as follows. Initially, Z visits $s(I)$ only, i.e., $Z = s(I)$. Repeatedly extend Z to visit the one with the leftmost right endpoint among neighbors of $\mathit{end}(Z)$ not visited by Z until all neighbors of $\mathit{end}(Z)$ are visited by Z. Then it outputs path Z and stops. For instance, given a set of 12 endpoint-sorted intervals shown in Fig. 1, Procedure **GP** outputs path $Z = i_1 \rightarrow i_2 \rightarrow i_3 \rightarrow i_4 \rightarrow i_6 \rightarrow i_5$.

For interval model I, define a path cover $PC(I)$ of $G(I)$ recursively as follows. If $I = \emptyset$, then $PC(I) = \emptyset$. Otherwise, let $PC(I) = \{Z\} \cup PC(I')$, where Z is the path obtained from I by Procedure **GP** and $I' = I - Z$. For instance, $PC(I) = \{Z_1 = i_1 \rightarrow i_2 \rightarrow i_3 \rightarrow i_4 \rightarrow i_6 \rightarrow i_5, Z_2 = i_7 \rightarrow i_8 \rightarrow i_9 \rightarrow i_{10} \rightarrow i_{12} \rightarrow i_{11}\}$, where I is the set of intervals shown in Fig. 1, Z_1 and Z_2 are the paths obtained by Procedure **GP** from I and $I - Z_1$, respectively.

Arikati and Rangan [1] proved by induction that $PC(I)$ is an optimal path cover of $G(I)$. Chang et al. [5] showed that $PC(I)$ can be computed in $O(n)$ time and $O(n)$ space given a set of n endpoint-sorted intervals.

Our certifying algorithm for the path cover problem on interval graphs works as follows. It computes $PC(I)$ as both an optimal path cover of $G(I)$ and a certificate showing that $PC(I)$ is indeed an optimal path cover of $G(I)$. Note that the length of $PC(I)$ is $O(n)$.

Before we show how to verify the optimality of $PC(I)$, let's observe the behavior of Procedure **GP** and the path obtained by it from I. It is easy to see that every interval in $I - \{s(I)\}$ is either a neighbor of $s(I)$ or to the right of $s(I)$. We see that Procedure **GP** maintains the following invariant while growing path Z to be output:

Every interval $x \in I - Z$ is either a neighbor of $end(Z)$ or to the right of $end(Z)$.

Initially, $Z = s(I)$ and the invariant holds trivially. Assume now $Z = z_1 \to z_2 \to \cdots \to z_h$ with $h \geqslant 1$ and the invariant holds. Let interval z be the one with the leftmost right endpoint among the neighbors of $end(Z)$ not in Z. There are two cases: either $z < z_h$ or $z_h < z$. Consider the former case first. By the variant, $z = s(I - Z)$. Hence every interval in $I - Z$ other than z is either a neighbor of z or to the right of z. Next we look into the later case. All neighbors of z_h not in Z are greater than z. For $x \in I - Z$ and $x < z_h$, x is a neighbor of z. For $x \in I - Z$ and $x > z$, x is either a neighbor of z or to the right of z. In other words, every interval in $I - Z$ other than z is either a neighbor of z or to the right of z. By the above observation we see that the invariant holds after the procedure extends Z to visit z in both cases. By induction then, the invariant maintained by Procedure **GP** holds. In addition, all intervals in $I - Z$ are to the right of $end(Z)$ for path Z output by Procedure **GP** by the invariant.

Let P be a path of $G(I)$. Define $L(P)$ to be the set of intervals in P which are larger than $end(P)$, i.e., $L(P) = \{x | x \in P \text{ and } end(P) < x\}$. Let

$$Z = z_1 \to z_2 \to \cdots \to z_k$$

be the path obtained by Procedure **GP** from interval model I. Notice that $x \neq z_1$ if $x \in L(Z)$. Recursively define $C(Z)$ as follows. If $L(Z)$ is the empty set, then $C(Z) = \emptyset$; Otherwise, let $C(Z) = C(Z') \cup \{z_i\}$, where i is the index such that $z_i \in L(Z)$ and $z_j \notin L(Z)$ for $i < j \leqslant k$ and $Z' = z_1 \to z_2 \to \cdots \to z_{i-1}$. For instance, let $Z = i_1 \to i_2 \to i_3 \to i_4 \to i_6 \to i_5$ be the path obtained by Procedure **GP** given the interval model shown in Fig. 1 and we have $C(Z) = \{i_6\}$.

Lemma 1. *Let $Z = z_1 \to z_2 \to \cdots \to z_k$ be the path obtained by Procedure* **GP** *from interval model I, i be the index such that $z_i \in L(Z)$ and $z_j \notin L(Z)$ for $i < j \leqslant k$ and $Z' = z_1 \to z_2 \to \cdots \to z_{i-1}$. Then, $I - Z$, $\{z_1, z_2, \ldots, z_{i-1}\}$, and $\{z_{i+1}, z_{i+2}, \ldots, z_k\}$ are islands with respect to $L(Z') \cup \{z_i\}$ in $G(I)$.*

Proof. By the invariant maintained by Procedure **GP**, intervals in $I - Z$ are to the right of z_k. Thus, $I - Z$ and $Z - L(Z)$ are islands with respect to $L(Z)$ in $G(I)$. By the greedy principle of Procedure **GP** and the invariant maintained by Procedure **GP**, all intervals in $\{z_{i+1}, z_{i+2}, \ldots, z_k\}$ are to the right of z_{i-1} and to the left of every interval in $I - Z$. Thus, $\{z_{i+1}, z_{i+2}, \ldots, z_k\}$ and $Z' - L(Z')$ are islands with respect to $L(Z') \cup \{z_i\}$ in $G(I)$.

The following corollary can be seen from the above lemma and the recursive definition of $C(Z)$.

Corollary 1. *Let Z be the path obtained by Procedure* **GP** *from interval model I, $C(Z) = \{c_1, c_2, \ldots, c_h\}$, and let $Z = Q_1 \to c_1 \to Q_2 \to c_2 \to \cdots \to Q_h \to c_h \to Q_{h+1}$. Then, $Q_1, Q_2, \ldots, Q_{h+1}$, and $I - Z$ are islands with respect to $C(Z)$ in $G(I)$.*

Suppose $PC(I) = \{Z_1, Z_2, \ldots, Z_q\}$, where $q = |PC(I)|$. Define $\mathbb{C}(I) = \cup_{1 \leqslant i \leqslant q} C(Z_i)$. For example, $\mathbb{C}(I) = \{i_6, i_{12}\}$ for the interval model I shown in Fig. 1, where $PC(I) = \{Z_1 = i_1 \to i_2 \to i_3 \to i_4 \to i_6 \to i_5, Z_2 = i_7 \to i_8 \to i_9 \to i_{10} \to i_{12} \to i_{11}\}$, $C(Z_1) = \{i_6\}$, and $C(Z_2) = \{i_{12}\}$. By Corollary 1, we have the following corollary:

Corollary 2. *There are $|PC(I)| + |\mathbb{C}(I)|$ connected components in $G(I - \mathbb{C}(I))$.*

By the above corollary and Proposition 1, we know that $PC(I)$ is an optimal path cover of $G(I)$. Now we are ready to describe our authentication algorithm as follows. Given $PC(I)$, the algorithm first checks whether $PC(I)$ is a path cover of $G(I)$ and then computes $\mathbb{C}(I)$ to test whether $G(I - \mathbb{C}(I))$ has $|PC(I)| + |\mathbb{C}(I)|$ connected components. Whether $PC(I)$ is a path cover of $G(I)$ can be checked in $O(n)$ time. Given $PC(I)$, $\mathbb{C}(I)$ can be computed in $O(n)$ time. Given a sorted interval model I and $\mathbb{C}(I)$, the number of connected components in $G(I - \mathbb{C}(I))$ can be determined in $O(n)$ time. Thus the running time of the authentication algorithm is $O(n)$ and we have the following theorem.

Theorem 2. *There is an $O(n)$-time certifying algorithm for the path cover problem on interval graphs given sorted interval models. The optimality of the output of this algorithm can be authenticated in $O(n)$ time.*

4 A Certifying Algorithm for the Hamiltonian Cycle Problem on Interval Graphs

In this section, we present a certifying algorithm to solve the Hamiltonian cycle problem on interval graphs. Given a sorted interval model, it runs in $O(n)$ time. It is assumed that a set I of n endpoint-sorted intervals is given and $n > 2$. The algorithm first computes path Z by calling Procedure **GP**. If $I - Z \neq \emptyset$ or $L(Z) \neq \emptyset$, then $G(I)$ has no Hamiltonian cycle. In this case, it outputs $C(Z)$ as a certificate. Clearly, $C(Z)$ is of length $O(n)$. The authentication algorithm simply checks whether $G(I - C(Z))$ has more than $|C(Z)|$ connected components. If $G(I - C(Z))$ does have more than $|C(Z)|$ connected components, then by Proposition 2 $G(I)$ does have no Hamiltonian cycle. Suppose $I - Z = \emptyset$ and $L(Z) = \emptyset$. That is, $G(I)$ has a Hamiltonian path Z with $L(Z) = \emptyset$. Then, we grow two paths P_1 and P_2 from Z. Let $Z = z_1 \to z_2 \to \cdots \to z_{n-1} \to z_n$. Initially, let $P_1 = z_1$ and $P_2 = z_2$. Then for i from 3 to n, we grow the two paths as follows. Let P^* be the one that visits z_{i-1} and P' be the other path. If z_i is adjacent to $end(P')$ then extend P' to visit z_i; otherwise extend P^* to visit z_i. After all intervals are visited by either one of the two paths, let P^* be the one that visits z_n and P' be the other path. We can easily see that $G(I)$ has a Hamiltonian

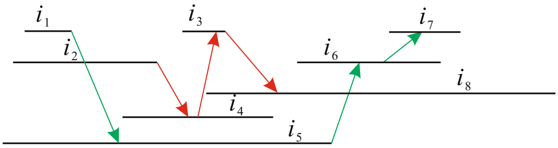


Fig. 2. A sorted interval model I with a Hamiltonian cycle, in which arrow lines indicate the visited sequences of the two paths

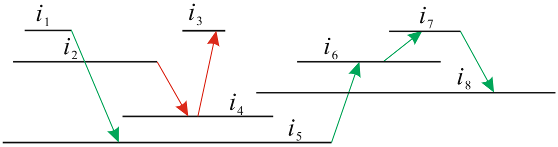


Fig. 3. A sorted interval model I without Hamiltonian cycle, in which arrow lines indicate the visited sequences of the two paths

cycle if P' visits z_{n-1} since z_1 and z_2 are adjacent and z_{n-1} and z_n are adjacent. In this case, the algorithm outputs the Hamiltonian cycle obtained from the two paths. On the other hand, it claims that $G(I)$ has no Hamiltonian cycle if P' does not visit z_{n-1}. Next, we show how to obtain a certificate proving that $G(I)$ has no Hamiltonian cycle in case that $z_{n-1} \notin P'$. Let h be the largest index such that $z_h \in P'$. Clearly, $h < n-1$. Let $Z' = z_1 \rightarrow z_2 \rightarrow \cdots \rightarrow z_{h-1} \rightarrow z_h$. By the above algorithm, $z_{h+1} \in P^*$. Then for $h+2 \leqslant i \leqslant n$, z_i is not adjacent to z_h by the algorithm and is to the right of z_h by the invariant maintained by Procedure **GP**. Notice that the set $\{z_{h+2}, \ldots, z_n\}$ is not empty. Hence, $\{z_{h+2}, \ldots, z_n\}$ is an island with respect to $L(Z') \cup \{z_{h+1}\}$ in $G(I)$. Then, there are at least $|C(Z')|+2$ islands with respect to $C(Z') \cup \{z_{h+1}\}$ in $G(I)$. The algorithm outputs $C(Z') \cup \{z_{h+1}\}$ as a certificate showing that $G(I)$ has no Hamiltonian cycle by Proposition 2.

For instance, given a set of sorted intervals shown in Fig. 2, $Z = i_1 \rightarrow i_2 \rightarrow i_4 \rightarrow i_3 \rightarrow i_5 \rightarrow i_6 \rightarrow i_7 \rightarrow i_8$. Then, $P_1 = i_1 \rightarrow i_5 \rightarrow i_6 \rightarrow i_7$ and $P_2 = i_2 \rightarrow i_4 \rightarrow i_3 \rightarrow i_8$. It is easy to see that a Hamiltonian cycle can be obtained from the two paths. On the other hand, given a set I of sorted intervals shown in Fig. 3, $Z = i_1 \rightarrow i_2 \rightarrow i_4 \rightarrow i_3 \rightarrow i_5 \rightarrow i_6 \rightarrow i_7 \rightarrow i_8$. Then, $P_1 = i_1 \rightarrow i_5 \rightarrow i_6 \rightarrow i_7 \rightarrow i_8$ and $P_2 = i_2 \rightarrow i_4 \rightarrow i_3$. We can see that $Z' = i_1 \rightarrow i_2 \rightarrow i_4 \rightarrow i_3$, $C(Z') = \{i_4\}$, and $z_{h+1} = i_5$. Then, $G(I - \{i_4, i_5\})$ has 3 connected components. By Proposition 2, $G(I)$ has no Hamiltonian cycle. We then have the following theorem.

Theorem 3. *There is an $O(n)$-time certifying algorithm for the Hamiltonian cycle problem on sorted interval models. The length of the certificate provided by the algorithm is $O(n)$ and the authentication algorithm runs in $O(n)$ time.*

5 A Certifying Algorithm for the Scattering Number Problem on Interval Graphs

Given a set I of n sorted intervals, we give an $O(n)$-time certifying algorithm to compute the scattering number $s(G(I))$ and provide a certificate for this answer as follows. By Theorem 1, if $G(I)$ has a Hamiltonian cycle, then $s(G(I)) \leqslant 0$; and if $G(I)$ has a Hamiltonian path, then $s(G(I)) \leqslant 1$. By Proposition 5, $\pi(G(I)) \geqslant \max\{1, s(G(I))\}$. Our certifying algorithm for computing $s(G(I))$ is then sketched as follows. It first calls the Hamiltonian cycle algorithm to test whether $G(I)$ has a Hamiltonian cycle. If it has a Hamiltonian cycle C, then $s(G(I)) = 0$ and C is provided as a certificate. Otherwise, the algorithm computes the minimum path cover $PC(I) = \{Z_1, Z_2, \cdots, Z_q\}$ of $G(I)$. If $q = 1$, i.e., $G(I)$ has a Hamiltonian path, then $s(G(I)) = 1$ and Z_1 is provided as a certificate. Otherwise, it computes $\mathbb{C}(I) = \cup_{1 \leqslant i \leqslant q} C(Z_i)$ and $\omega(G(I) - \mathbb{C}(I))$. It then outputs $s(G(I)) = \omega(G(I) - \mathbb{C}(I)) - |\mathbb{C}(I)| = |PC(I)| = \pi(G(I))$, and provides $\mathbb{C}(I)$ as a certificate. For example, given a set I of sorted intervals, as shown in Fig. 1. The algorithm first computes $PC(I) = \{Z_1 = i_1 \rightarrow i_2 \rightarrow i_3 \rightarrow i_4 \rightarrow i_6 \rightarrow i_5, Z_2 = i_7 \rightarrow i_8 \rightarrow i_9 \rightarrow i_{10} \rightarrow i_{12} \rightarrow i_{11}\}$, $C(Z_1) = \{i_6\}$, and $C(Z_2) = \{i_{12}\}$. Then, $\mathbb{C}(I) = C(Z_1) \cup C(Z_2) = \{i_6, i_{12}\}$ and $\omega(G(I) - \mathbb{C}(I)) = |PC(I)| + |\mathbb{C}(I)| = 2 + 2 = 4$. The algorithm then outputs $s(G(I)) = \omega(G(I) - \mathbb{C}(I)) - |\mathbb{C}(I)| = 2$, and provides $\mathbb{C}(I)$ as a certificate. The correctness of the above algorithm follows from Proposition 3, Corollary 2, and Theorem 1. By Theorems 2 and 3, the above algorithm runs in $O(n)$ time. It is easily seen that the provided certificates can be authenticated in $O(n)$ time. Thus the running time of the authentication algorithm is $O(n)$ and we conclude the following theorem.

Theorem 4. *There is an $O(n)$-time certifying algorithm for the scattering number problem on interval graphs given sorted interval models. The length of the certificate provided by the algorithm is $O(n)$ and the authentication algorithm runs in $O(n)$ time.*

References

1. Arikati, S.R., Pandu Rangan, C.: Linear Algorithm for Optimal Path Cover Problem on Interval Graphs. Inform. Process. Lett. 35, 149–153 (1990)
2. Ascheuer, N.: Hamiltonian Path Problems in the On-Line Optimization of Flexible Manufacturing Systems. Technique Report TR 96-3, Konrad-Zuse-Zentrum für Informationstechnik, Berlin (1996)
3. Boesch, F.T., Gimpel, J.F.: Covering the Points a Digraph with Point-Disjoint Paths and its Application to Code Optimization. J. ACM 24, 192–198 (1977)
4. Bondy, J.A., Murty, U.S.R.: Graph Theory with Applications, New York (1976)
5. Chang, M.S., Peng, S.L., Liaw, J.L.: Deferred-Query: an Efficient Approach for Some Problems on Interval Graphs. Networks 34, 1–10 (1999)
6. Chu, F.P.M.: A Simple Linear Time Certifying LBFS-Based Algorithm for Recognizing Trivially Perfect Graphs and their Complements. Inform. Process. Lett. 107, 7–12 (2008)

7. Chvátal, V.: Tough Graphs and Hamiltonian Circuits. Discrete Math. 5, 215–228 (1973)
8. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
9. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Annals of Discrete Mathematics 57 (2004)
10. Heggernes, P., Kratsch, D.: Linear-Time Certifying Algorithms for Recognizing Split Graphs and Related Graph Classes. Nordic J. Comput. 14, 87–108 (2007)
11. Hell, P., Huang, J.: Certifying LexBFS Recognition Algorithms for Proper Interval Graphs and Proper Interval Bigraphs. SIAM J. Discrete Math. 18, 554–570 (2005)
12. Jung, H.A.: On a Class of Posets and the Corresponding Comparability Graphs. J. Combin. Theory Ser. B 24, 125–133 (1978)
13. Kratsch, D., McConnell, R.M., Mehlhorn, K., Spinrad, J.P.: Certifying Algorithms for Recognizing Interval Graphs and Permutation Graphs. In: Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2003), pp. 158–167 (2003)
14. Kratsch, D., McConnell, R.M., Mehlhorn, K., Spinrad, J.P.: Certifying Algorithms for Recognizing Interval Graphs and Permutation graphs. SIAM J. Comput. 36, 326–353 (2006)
15. Manacher, G.K., Mankus, T.A., Smith, C.J.: An Optimum $\Theta(n \log n)$ Algorithm for Finding a Canonical Hamiltonian Path and a Canonical Hamiltonian Circuit in a Set of Intervals. Inform. Process. Lett. 35, 205–211 (1990)
16. Ntafos, S.C., Louis Hakimi, S.: On Path Cover Problems in Digraphs and Applications to Program Testing. IEEE Trans. Software Engrg. 5, 520–529 (1979)
17. Pinter, S., Wolfstahl, Y.: On Mapping Processes to Processors. Internat. J. Parallel Programming 16, 1–15 (1987)
18. Shih, W.K., Chern, T.C., Hsu, W.L.: An $O(n^2 \log n)$ Time Algorithm for the Hamiltonian Cycle Problem on Circular-Arc Graphs. SIAM J. Comput. 21, 1026–1046 (1992)
19. Toussaint, G.T.: Pattern Recognition and Geometrical Complexity. In: Proceedings of the 5th International Conference on Pattern Recognition, Miami Beach, pp. 1324–1347 (1980)
20. Waterman, M.S., Griggs, J.R.: Interval Graphs and Maps of DNA. Bull. Math. Biol. 48, 189–195 (1986)

Design and Overhead Estimation of Device Driver Process

Yusuke Nomura[1], Kouta Okamoto[1], Yusuke Gotoh[1], Yoshinari Nomura[1],
Hideo Taniguchi[1], Kazutoshi Yokoyama[2], and Katsumi Maruyama[3]

[1] Graduate School of Natural Science and Technology,
Okayama University, Okayama, Japan
{yusuke,okamoto}@swlab.cs.okayama-u.ac.jp,
{gotoh,nom,tani}@cs.okayama-u.ac.jp
[2] Research and Development Headquarters, NTT DATA Co., Tokyo, Japan
yokoyamakz@nttdata.co.jp
[3] National Institute of Informatics, Tokyo, Japan
maruyama@research.nii.ac.jp

Abstract. Conventional operating systems used to have device drivers as kernel modules or embedded objects. Therefore, maturity of a device driver influences on the reliability of the entire system. There is a method for constructing device driver as an user process for improving the reliability. Device driver process enhances the reliability of the operating system. However, device driver process has large overhead. In this paper, we propose a method for constructing device drivers process and evaluating these overhead. Also, this paper shows that the overhead of device driver process can be estimated.

Keywords: Operating system, Device driver, Microkernel, Overhead.

1 Introduction

Due to the rise of computer technology, many kinds of devices appear. In operating system (OS), a device is controlled by the program called a device driver. In order to make a new device for supporting OS, it is necessary to develop the device driver. The reliability of the device driver is different according to a skill of developers.

In conventional operating systems, the device driver is embedded in a kernel. When a trouble of the device driver occurs, since the computer system may be shut down, the reliability of it greatly affects that of the OS. The results [1] show that most parts of troubles in Linux occurs in the device driver. In Windows 2000, the results [2] show that the trouble in the kernel is 2% of the total, and that of the device driver is 27% of the total. Therefore, the improvement of the reliability in the device driver is important for the reliability of OS.

In order to improve the reliability of OS, several methods for separating the device driver from the kernel as a user process are proposed [3,4,5]. These methods can limit harmful effects by making the device driver as the process

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 324–335, 2010.

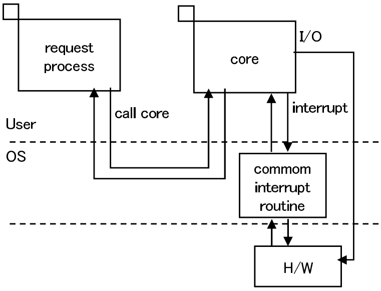


Fig. 1. Device driver process

that runs in high independency. For example, when the device driver is stopped, since the process can be executed continuously by activating another driver, the reliability of the system is improved. When a harmful effect of the device driver occurs, the influence on the OS and other processes becomes smaller by making the process. However, these methods execute the system call and require the process in executing I/O instruction, the overhead of executing the system call is large. Moreover, when the device driver executes the process, these methods do not explain about that the overhead of switching it occurs.

In this paper, we propose an efficient method for the device driver process. Our proposed method improves the reliability of OS by controlling the frize of OS. We explain about the overhead of the process and evaluate it. Also, in our proposed method, by using the mechanism of changing a running mode [6], the number of system call estimation becomes reduced. Therefore, the overhead of the system call emission can be reduced.

This paper is organized as follows. We explain the device driver process in Section 2. Our proposed method is explained in Section 3, and evaluated in Section 4. Related works are introduced in Section 5. Finally, we conclude the paper in Section 6.

2 Process Model of Device Driver

2.1 Basic Idea

The outline of device driver process is shown in Figure 1. The device driver which is realized as the process is called *core*.

The basic process of the core is explained below.

1. When the request process is required, the core is executed by switching the process in the kernel.
2. At interrupt, the interrupt processing in the core is called directly.
3. In I/O control of the hardware, the core is set directly without the kernel.

For example, it explains the case of the reading demand for the disk driver. When the request process is required to the disk driver, the disk driver is excuted by switching the process in the kernel. The disk driver issues the reading instruction to the disk. The disk driver receives the interrupt of the reading completion. When the disk driver's processing is finished, the request process continues processing again.

2.2 Mechanism of Changing Running Mode for Application Program

In our system, we use a mechanism of changing the running mode for the application program. The running mode of the process is changed from user mode to supervisor mode or otherwise. Since the process in supervisor mode can call the system call of OS in the form of function call, the overhead of system call can be reduced. By running the device process in supervisor mode using this mechanism, since the overhead can be reduced by the system call in I/O instruction, the device driver is executed effectively.

However, when our system use the mechanism of changing the running mode, the protection of kernel area in running the process of supervisor mode becomes low. In our system, two different virtual address spaces of protection information in the kernel data area for one process space are provided. When the running process in supervisor mode run in the user space, by setting the data area of the kernel for only reading data, it prevents information alteration from the kernel data area accidentally.

2.3 Requirements

Issues of realizing the basic idea are below.

1. Determining the core of calling in target
   When the process requires the processing request, information of specifying the core is needed. In our system, we use the 32 bit core ID for specifying the core. A core ID format is shown in Figure 2.
2. Data Communication between processes
   The requiring process and the sending core it is made as a different process. In data communication such as a parameter and a return value, sending data between different virtual addressing spaces is needed.
3. Detection of finishing cell process
   The requiring processing is waits until the core of sending it is finished. The kernel needs to detect finishing the cell process to awake the waiting state of the process. In our system, the kernel receives information about finishing the core process or not as a return value of processes in system call and interrupt.
4. Interrupt process
   In processing of using interrupt, the kernel calls the interrupt processing of the core in relation to the number of interrupt. However, since the core is operated as a process, the kernel can not call the interrupt processing directly.

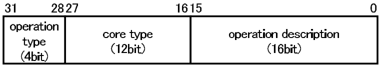


Fig. 2. Core ID format

2.4 Our Solutions

For issues explained in Subsection 2.3, we present two methods below.

1. Our system waits until the core gets an execute authority in interrupt and executes the interrupt process in the context of the core.
2. Our system switches virtual address space of the process to that of the core in interrupt and calls the interrupt procedure. This process is called an immediate interrupt.

In Method 1, since the context of the core is switched in interrupt, waiting time occurs until executing the interrupt. The core needs to manage the interrupt when multiple interrupt occurs before getting the execute authority, and calls it in the context. Therefore, the process becomes complex. However, since the core is operated in the process control of OS, the influence on other processes is diminished.

In Method 2, since the interrupt process is executed by switching the virtual address space immediately in interrupt, the delay from occurring interrupt to executing it is small. Also, since interrupt process is called immediately using function call in interrupt, the process becomes simple. However, TLB flush is executed in switching the virtual address space. After switching the virtual address space and executing the interrupt, TLB miss hit occurs in the running process before interrupt. In this case, the system performance becomes low.

In our paper, we use the Method 2 considering the currency in interrupt and the simplification of the process.

3 Proposed Method

The device driver that operates depending on the kernel is called an embedded driver. Also, the process device driver is called a process type driver. The device driver needs to call the OS function frequently to wakeup the process (starting process) and sleep the event (waiting event). In our proposed method, the process type driver curbs the rise in the number of OS function call by operating within the kernel without calling the starting process and the waiting event. An expected effectiveness by constructing the process type driver is explained below.

1. Improving the reliability of OS
   In the case of the embedded driver, when the failure occurs in the device driver, OS may be effected. Otherwise, in the case of the process type driver, when the failure occurs in the device driver, the effectiveness of OS can be diminished.

2. Work improvement
   Since the process type driver can be implemented as well as the process of the application program (AP) with providing the service, the work of the device driver may be improved. Since the development method of AP easily applies to the device driver in a developmental stage, we can develop a high quality device driver easily.
3. Effective I/O
   Since the process type driver can run in supervisor mode by constructing of changing the running mode, it can reduce the overhead of system calls in I/O instructions, and the device driver is used effectively. Since the process using supervisor mode is not effected by the protection of running mode of OS, it can issue the instruction in supervisor mode, and read and write data to OS area.

3.1 Overhead for Constructing Device Driver Process

Items
The process type driver has several types of the overhead compared to the embedded driver. Factors for the overhead are shown below.

1. Increasing the number of dispatching process
   Since the process type driver operates as being different from the process that requires the process to the driver, the number of process switching increases compared to the embedded driver. The overhead of process switching includes process time in OS scheduling process and the virtual address space switching. This factor will be evaluated in subsection 4.1.
2. Increasing the number of dispatching virtual address space
   The process switching includes the virtual address space switching. The process type driver may operate the virtual address space switching before the interrupt. For example, in the case of using Pentium processor, Translation Lookaside Buffer (TLB) flush occurs in the address address space switching. Therefore, it would appear that the process type driver is affected by TLB flush. This factor will be evaluated in subsection 4.2.
3. Increasing the number of overhead for I/O instructions
   The device driver needs to operate I/O instructions for controlling the machine. However, since I/O is a supervisor instruction, the process type driver requires operating I/O instructions to OS by the system call. This factor will be evaluated in subsection 4.3.

Analysis
We compare the process type driver with the embedded driver and evaluate the wrong effect by the device driver process.

1. Increasing the number of dispatching process
   In the process type driver, situations of the process switching in cases of

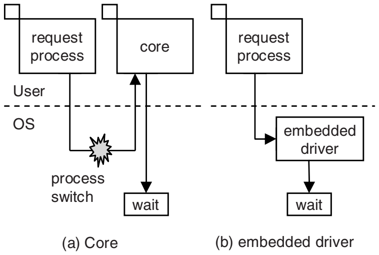


Fig. 3. Process switching in callcore

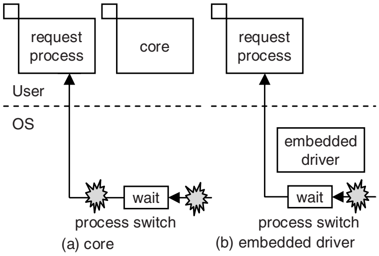


Fig. 4. Process switching in returning from callcore

requiring process and returning from it are shown in Figures 3 and 4. In Figure 3, one process switching occurs before the process requires the process to the core and executes the system call. On the other hand, in the embedded driver, the process is changed to the driver without process switching. In Figure 4, when the system call is finished and the waiting state of the core dissolve by interrupt, one process switching occurs. After the core executes the process, one process switching occurs when the requiring process is returned. In the embedded driver, one process switching occurs only when the waiting state of the process dissolve. Therefore, in the process type driver, the number of switching process increases twice compared to the embedded driver in requiring the process.

2. Increasing the number of dispatching virtual address space
   In the process driver, the interrupt process is called by switching the virtual address space of the core to virtual address space of the device driver temporarily in interrupt. In Figure 5, the number of switching the virtual address space increases twice compared to the embedded driver. Therefore, switching the virtual address space is up to two and zero at minimum compared to the embedded driver in the core.
3. Increasing the number of overhead for I/O instructions
   The core in user mode requires I/O instruction to OS using system call. Therefore, the overhead of I/O instruction depends on process time of system call and the number of I/O instruction of the device driver.

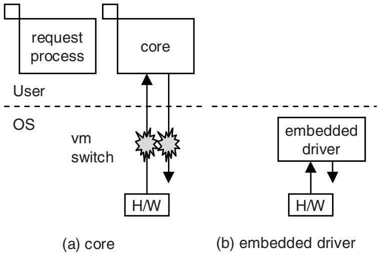


Fig. 5. Switching virtual address space in interrupt

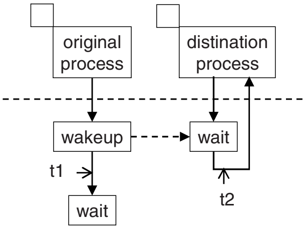


Fig. 6. Measurement of process switching

Table 1. Result of process switching time

Average	9777
Max	12004
Min	9612

4 Evaluation

Here, we evaluated a basic overhead of our proposed method using OS in FreeBSD 4.3R and CPU in Pentium 4 2.8GHz. Process time is measured using a *rdtsc* instruction, and a measured result is the average number of clocks for 10,000 times. In measuring the number of TLB hit, we use a *rdpmc* instruction.

4.1 Process-Switching-Time

Evaluation environment is shown in Figure 6. In Figure 6, we evaluated process switching time between t_1 and t_2. By placing a high priority on an original process, we get a destination process to work next to it in OS scheduling. An evaluation result is shown in Table 4.1. In Table 4.1, average process switching time is about 9,800 clocks.

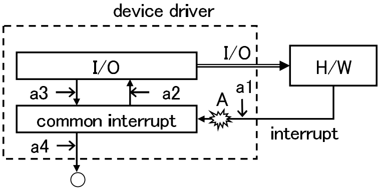


Fig. 7. Measurement of interrupt handler

Table 2. Result of measurement in interrupt

Switching memory	Average	Max	Min	TLB miss
No	13644	15435	13345	0
Yes	14021	14711	13692	0
Difference	377	-	-	0

4.2 Address Space-Switching Time

We evaluated virtual address space-witching time by switching the process using the embedded driver. In Pentium 4 processor, the virtual address space-switching is operated by setting the initial address of a page directory in CR3 register. We simulated the virtual address space-switching process by reconfiguring the same value to CR3 register.

A measurement of interrupt handler in NIC is shown in Figure 7. The interrupt handler changes the setting of allowing interrupt in NIC and verifies the state of it. We evaluated processing time for an interrupt call of the driver process in each case of operating and not operating the virtual address space switching process of point A in Figure 7. In Figure 7, since the I/O process (between a2 and a3) has a large margin of process time, we eliminated it from the measurement. In our evaluation, we measured total process time between a1 and a2, and between a3 and a4.

An evaluation result is shown in Table 2. In Table 2, average time of switching virtual address space is about 380 clocks.

Next, we evaluated TLB miss hit time. To measure the influence of TLB miss hit, reading data of users is used in sending data of NIC driver that is implemented as an embedded driver. Since user data is in its area, TLB miss hit occurs in reading data.

A measurement situation of sending data in NIC is shown in Figure 8. In sending data of NIC driver, a memory address that stores data is written to the NIC register. Assuming to switch the virtual address space in process switching, in point B of Figure 8, we measure TLB miss hit time, which occurs in switching virtual address space. To measure only the influence of TLB flush, we measure time in reading data between b2 and b3. Data size of measuring time in reading data is 4 bytes.

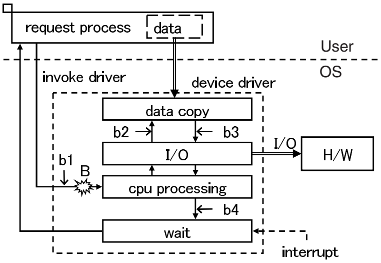


Fig. 8. Measurement of sending data

Table 3. Result of data read time

Switching memory	Average	Max	Min	TLB miss
No	416	557	388	0
Yes	518	599	472	2
Difference	102	-	-	2

Table 4. Processing time for NIC driver

Process	Processing time (clock)
Sending	6897
Interrupt	13644

A measurement result is shown in Table 3. In reading data, TLB miss hit occurs two times, which the delay of their process time is about 102 clocks. Therefore, we confirmed that the average delay of TLB miss hit is about $102/2 = 51$ clocks.

4.3 Overhead of I/O Instructions

In supervisor mode, since the system does not require the process to OS in I/O instruction, the overhead is not occurred. On the other hand, in user mode, since the system requires the process to OS by system call in I/O instruction, the overhead occurs. We estimated the overhead in I/O instruction based on the number of I/O instruction in the device driver and the overhead in executing the system call.

In case of requiring sending data of 4 byte to NIC driver, each processing time of sending process and interrupt for NIC driver is shown in Table 4.3, and the number of I/O instruction for NIC driver is shown in Table 5. Also, processing time of I/O instruction for each running mode is shown in Table 6.

In Table 6, the overhead for executing system call is about 900 clocks. In Tables 4.3 and 5, in user mode, the overhead of sending process is about 20%

Table 5. Number of I/O instruction for NIC driver

Process	I/O instruction
Sending	2
Interrupt	3

Table 6. Processing time for I/O instruction

Running mode	IN instruction	OUT instruction
User mode	3629	3796
Supervisor mode	2708	2855
Difference	921	941

and that of interruption is about 14 %. We confirmed that the operation of changing running mode has a high efficiency.

4.4 Discussion

When T_{sys} is the overhead in the system call and T_{intr} is the interrupt overhead, they are calculated below. t_{proc} is process switching time, t_{vm} is processing time of switching virtual address space, t_{tlb} is processing time of TLB miss, n_{pg} is the number of program pages, and N_{vm} is the number of switching virtual address space.

$$T_{sys} = t_{proc} \times 2 + t_{tlb} \times n_{pg} \tag{1}$$

$$T_{intr} = (t_{vm} + t_{tlb} \times n_{pg}) \times N_{vm} \tag{2}$$

In Subsections 4.1 and 4.2, since $t_{proc} = 9,777$, $t_{vm} = 377$, and $t_{tlb} = 51$, the increasing value of processing time can be calculated by n_{pg} and N_{vm}. Also, by reducing the number of pages that the program is used and switching the virtual address space, we confirmed that the overhead of making the process can be reduced.

5 Related Works

In MINIX 3 [3], User-level device driver, and Prime, the device driver is worked as a process to improve the reliability of OS. Since MINIX 3 does not use a paging, virtual address space switching is not occurred in requiring the device driver and interrupt. Therefore, MINIX 3 has a good effectiveness. However, in I/O instruction, since the process is required to the kernel by system call, the overhead of changing the running mode occurs. In User-level device driver and Prime, since each communication becomes high by sharing memory between the process and the device driver, and between OS and the device driver, the performance of it is close to that of conventional Linux drivers. Otherwise, the discussion about the safety of the device driver is not sufficient.

In our proposed method, the device driver that has a high reliability can be worked in supervisor mode using the mechanism of changing the running mode. When the process calls OS directly, since it can execute I/O instruction and memory mapped I/O, the overhead of changing the mode in system call can be reduced. Therefore, our proposed method is more effective than conventional methods in executing the process. On the other hand, when the reliability of the device driver is low, it can keep the safety by executing it in user mode. When data is received and sent between the process and the device driver, since it is copied, the overhead of communication becomes large. A method to reduce the overhead of communication is a future work.

Nooks [7] makes the safe execution environment for the device driver in the kernel. The kernel is protected by verifying the transfer of the data between the device driver and other modules in the kernel. The state of the device driver can be recovered by preserving the state of the memory.

SPIN [8] can dynamically extend the function of OS. Because the extension is linked with OS by a small overhead, it is possible to cooperate with OS. Since the extension is described Modula-3, the safety of OS is improved.

On the other hand, we can apply a past process observation method to the core because the core is executed as a process. In this case, execution, stopping, and the correction of the core become easy.

6 Conclusion

In our paper, we proposed the method for composing the device driver as a process to improve the reliability of OS. Also, we explained about the requirement, the solution of our proposed method, and the mechanism of the device driver. Our proposed method has the mechanism of changing running mode. When the device driver is developed and the reliability is low, the system is executed in user mode. On the other hand, a proven device driver is executed in supervisor mode. Therefore, device driver process has the trade-off of the performance and reliability. Our system can select the improving the reliability of the device driver.

In the future, we will make a faster connection between the process and the core and evaluate the system using the application program. Also, we will confirm that the evaluation basis is reliable.

References

1. Chou, A., Yang, J., Chelf, B., Hallem, S., Engler, D.: An empirical Study of Operating Systems Errors. In: Symposium on Operating Systems Principles, pp. 73–88 (2001)
2. Murphy, B.: Fault Tolerance role in this high availability world, http://research.microsoft.com/bmurphy/Fault%20Tolerance_files/frame.htm

3. Tanenbaum, A.S., Woodhull, A.S.: Operating Systems Design And Implementation, 3rd edn. Prentice-Hall, Englewood Cliffs
4. Leslie, B., Chubb, P., Fitzroy-Dale, N., Goetz, S., Gray, C., Macpherson, L., Potts, D., Shen, Y., Elphinstone, K., Heiser, G.: Userlevel Device Drivers: Achieved Performance. Journal of Computer Science and Technology 20, 654–664 (2005)
5. Elphinstone, K., Goetz, S.: Initial Evaluation of a User-Level Device Driver Framework. In: Proceedings of the 9th Asia-Pacific Computer Systems Architecture Conference, pp. 256–269 (2004)
6. Yokoyama, K., Nomura, Y., Taniguchi, H., Maruyama, K.: Process control mechanism for dynamic running mode switch of application program. The IEICE Transactions on Information and Systems J91-D(3), 696–708 (2008)
7. Swift, M.M., Martin, S., Levy, H.M., Eggers, S.J.: Nooks: An Architecture for Reliable Device Drivers. In: Proceedings of the Tenth ACM SIGOPS European Workshop (2002)
8. Bershad, B.N., Savage, S., Pardyak, P., Sirer, E.G., Fiuczynski, M.E., Chambers, C., Becker, D., Eggers, S.: Extensibility, Safety and Performance in the SPIN Operating System. In: Proceedings of 15th ACM Symposium on Operating system Principles, pp. 267–284 (1995)

Improvement of Gaze Estimation Robustness Using Pupil Knowledge

Kohei Arai[1] and Ronny Mardiyanto[1,2]

[1] Depatment of Information Science,
Saga University, Japan
[2] Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
arai@is.saga-u.ac.jp,ronny_mardiyanto@yahoo.com

Abstract. This paper presents an eye gaze estimation system which robust against various users. Our method utilizes an IR camera mounted on glass to allow user's movement. Pupil knowledge such as shape, size, location, and motion are used. This knowledge works based on the knowledge priority. Pupil appearance such as size, color, and shape are used as the first priority. When this step fails, then pupil is estimated based on its location as second priority. When all steps fail, then we estimate pupil based on its motion as the last priority. The aim of this proposed method is to make the system compatible for various user as well as to overcome problem associated with illumination changes and user movement. The proposed system is tested using several users with various race as well as nationality and the experiment result are compared to the well-known adaptive threshold method and template matching method. The proposed method shows good performance, robustness, accuracy and stability against illumination changes without any prior calibration.

Keywords: Gaze, eye detection, pupil, pupil knowledge.

1 Introduction

The method for gaze estimation is divided into the followings by the method for eye detection and tracking with line of sight vector estimation [1], [2], [3], [4];

1. Limbus Tracking method [5] which uses a reflectance difference between sclera: white-of-the-eye and cornea. Key issue here is how to find black-of-the-eye in particular, iris-of-the-eye (iris) and track the rims of the sclera and cornea.
2. Purkinje method [6]: Using the near-infrared wavelength-band reflected figure from the field where the refractive indices of the front of a cornea, the rear surface, an anterior surface of the lens, and the rear surface differ. Key issue here is how to use the image with the brightest luminosity of a Purkinje image
3. EOG (Electro-occulo-graphy) method [7] uses voltage differences between fore and aft surface of eyes. Key issue here is how to stick an electrode near

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 336–350, 2010.

an eyeball and measure the potential difference of + potential in a cornea, and - potential in the retina

4. Search coil method [8] uses generating voltage with coil including in contact lenses attached to user's eyes. Key issue here is how to place the user putting on the contact lens having a coil into a magnetic field, and to use the potential difference to generate with the angle of a coil and a magnetic field by eye movement
5. Fundus Haploscope [9] uses bottom surface images of user eyes with infrared cameras. Key issue is how to carry out direct observation of the fundus of the eye using the fundus-of-the-eye photograph-imaging device in an infrared wavelength band. The eye's fundus is the only part of the human body where the micro circulation can be observed directly.
6. Image analysis method [10], [11], [12], [13], [14], [15] uses user's eyes images acquired with cameras through eye detection and tracking. Key issue here is how to acquire pictures of eyeball and to detect a pupil center based on an image processing and analysis method. An eyeball is acquired and picturized with cameras and there is a method of detecting a pupil center by an image processing and analysis method etc.

The methods (1) to (5) do not impose users any psychological and physical burden because the methods require direct sensors attached to user's face. Also the methods are relatively costly. On the other hand, the image analysis method does not insist any load to users and is realized with comparatively cheap system. For the aforementioned reason, we propose a method based on the image analysis method.

In gaze estimation based on image analysis [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27] almost all utilize pupil location as reference. Accuracy of pupil detection was very important because all the gaze calculation are made based on this value. Many researcher did not give much intention on this part because most of them use ideal images as the source in order to find pupil location. Ideal eye images which look at forward has clearly pupil. It is very easy to find the pupil on this image type. Unfortunately, the real circumstances have a lot of various due to their shape and others. Although the appearance of pupil same at each users, change of eye shape when eye moves make pupil appearance will change following this changes. Also, when users have thick eyebrows, this will to be disruption when analyzing pupil. Varies of skin color, race, interference with eyelids, disappearances, and changes of eye shape (when eye move) are the reasons that make the pupil detection very difficult.

In this paper, we proposed gaze estimation system, which is robust for various users and also allows user movement, illumination changes, and vibration. Our system utilizes IR camera mounted on glass to allow user‘s movement. Pupil knowledge such as shape, size, location, and motion are used. This knowledge works based on the knowledge priority. Pupil appearance such as size, color, and shape is used as first priority. When this step fails, then pupil is estimated based on its location as second priority. When all steps fail, then we estimate pupil based on its motion as last priority. The objective of the proposed method is to overcome varies user, illumination changes, and user movement problem of previous methods. The proposed system is tested using several user with varies race and nationality and the experiment result are compared to well-known adaptive threshold method.

This paper is organized as follows: section 2 describes related work, section 3 explains the proposed system, section 4 presents the experimental results, section 5 describes the results discussion, and the rest concludes the paper.

2 Related Work

The published pupil detection approaches can be broadly classified into two categories: the active infrared (IR)-based approaches [17], [18], [26] and the traditional image-based passive approaches [19], [20], [21], [22], [23], [24], [25]. Eye detection based on Hough transform is proposed [19], [20], [21]. Hough transform is used in order to find the pupil. Eye detection based on motion analysis is proposed [16], [17]. Infrared lighting is used to capture the physiological properties of eyes (physical properties of pupils along with their dynamics and appearance to extract regions with eyes). Motion analysis such as Kalman filter and Mean Shift tracking are used. Support vector machine classifier is used for pupil verification. Eye detection using adaptive threshold and Morphologic filter is proposed [22]. Morphologic filter is used to eliminate undesired candidates for an eye. Hybrid eye detection using combination between color, edge and illumination is proposed [25].

Eye detection based on motion analysis [17], [18] will fail when eyes are closed or occluded. In our system, we cannot use this method. Our pupil detection have to works when eye shape changes. In first running, after adaptive threshold step, such ambiguity problem will appear because of error of threshold value. The threshold error is caused by threshold value which always change following the image condition that ultimately makes the threshold value very hard to be defined perfectly. Ambiguity arise when detection is done by considering their movement only. When eyeball moves, eyebrow and pupil will also move and yield two kinds of motion. Because of other eye components move when eye moves, the ambiguity problem to distinguish between eyebrow and pupil will happen. Eye detection based on Hough transform [19], [20], [21] is not robust against noise influence, pupil detection based on Hough transform also has less success rate. Deformable pupil shape also influence the result other than noise. We will be faced with robustness problem when Hough transform is used, because it only rely on their shapes. Eye detection using morphological filter [22] which eliminate noise and undesired candidate of eye will not robust against user variance. Morphologic method will not work when noises have same shape and size with pupil. Eye detection based on template matching [23], [24], segments of an input image are compared to previously stored images, to evaluate the similarity of the counterpart using correlation values.

The problem with simple template matching is that it cannot deal with eye variations in scale, expression, rotation and illumination. Use of multi scale templates was somewhat helpful in solving the previous problem in template matching. When eye detector only relies on eye appearance [25], this method will fail when eye unseen or closed. This method also will be faced on variance user color skin.

In this paper, we propose robust pupil detection using adaptive threshold and pupil knowledge. In order to extract pupil, active IR illumination based approach is used. This approach is robust against illumination changes and does works when environment light source is not available. It utilizes the special bright pupil effect under IR to detect and track the eyes. Several active IR based eye trackers [17], [18], [26] were proposed, but most of them require distinctive bright pupil effect to work well because they all track the eyes by tracking the bright pupils. The success of such a system strongly depends on the brightness and size of the pupils, which are often function of eye closure, face orientations, external illumination interferences, and the distances of the subjects to the camera. The common problems of pupil detection using IR camera are pupil ambiguity and robustness against different user. When adaptive threshold is applied onto eye image, there are components which are similar with pupil. Our proposed method use pupil knowledge to estimate its location. After adaptive threshold is applied into image, then pupil knowledge such size, shape, location, and motion is used. We distinguish between pupil and the components which are similar with pupil by priorities. The first priority is size and shape. If there is component which has size and shape looks like pupil, then we judge that this is pupil. When there are components look like pupil more than one, we choose that the closest pupil with previous location is correct pupil. Last case is, when there are candidates pupil which have same size, shape, and location, we estimate the correct pupil by analyzing its motion. We compared our result with the adaptive threshold method and template matching method using various user who have different race and nationality.

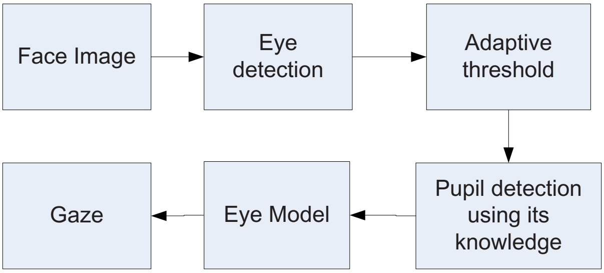


Fig. 1. Proposed gaze estimation block diagram. After face image is acquired using camera, eye detection is used to search roughly position of eye. Next, the detail position is searched based on pupil location. Gaze value is obtained by using eye model and pupil location.

3 Proposed System

3.1 Hardware Configuration

Optiplex 755 dell computer with Core 2 Quad 2.66 GHz CPU and 2G RAM is used. We develop our software under C++ Visual Studio 2005 and OpenCv Image processing Library which can be downloaded as free on their website. Our system utilize infrared web camera NetCowBow DC-NCR 131 as real time data input. This camera has benefit when illumination changes. By using 7 IR LED, this camera will adjust the illumination. This will cause the input image has minimum influence against illumination changes. The camera is mounted on the glass. When user head move, camera automatically follows because it attached on user glass. Also, when vibration happen, user body will reduce vibration effect automatically. The distance between camera and eye is 15.5 cm. Our gaze estimation hardware is shown in Fig. 2.

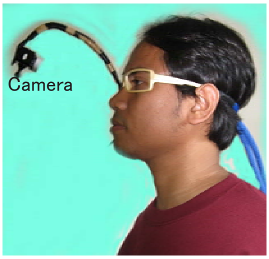


Fig. 2. Gaze Estimation Hardware. This figure shows that when user move, camera also moves following head movement.

3.2 Gaze Estimation

In order to analyze eye gaze, eye should be detected and tracked first. Fig. 3 shows flow of eye detection and tracking. Typically, our system detects eye based on deformable template method [27]. This method matches between eye template and source images. We create eye template by apply Gaussian smother onto this image. Deformable template method detects roughly position of eye. Benefit of deformable template method is that it takes less time than classifier methods. Although this method faster than classifier method, the robustness still less. In our system, when deformable template fails to detect eye position, viola-Jones classifier will detects eye. It means that Viola-Jones method is used only when deformable template fails to detect eye. The viola-Jones classifier employs adaboost at each node in the cascade to learn a high detection rate the cost of low rejection rate multi-tree classifier at each node of the cascade. To apply the viola-Jones classifier onto system, we use viola-Jones function in OpenCV [16]. Before

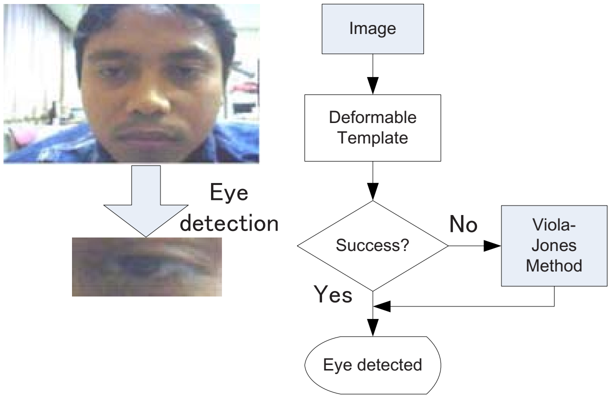


Fig. 3. Eye detection and tracking. When deformable template method fails, viola-jones method will take over to search eye.

use the function, we should create XML file. The training samples (face or eye image) must be collected. There are two samples: negative and positive sample. Negative sample corresponds to non-object images. Positive sample corresponds to object image. After acquisition of image, OpenCV will search the face center location followed by searching the eye center location. By using combination between deformable eye template and Viola-Jones method, eye location is detected. Advantages of these methods are fast and robust against circumstances change. After the roughly eye position is found, eye image is locked and cropped based on this position. It means that we should not repeat the eye detection again. The position of eye will not be changes because the camera is mounted on the glass. Eye gaze is estimated based on pupil location. Because of this system rely on the pupil location, pupil detection with perfectly accurate and robustness is required. Pupil is detected by using its knowledge. Flow of pupil detection is shown in Fig. 4. Three types of knowledge are used. We use pupil size, shape, and color as the first knowledge. First, adaptive threshold method is applied onto eye image. Threshold value T is obtained from average pixel value (mean) of eye image μ. We set threshold value is 27% bellow from mean.

$$\mu = \frac{1}{N}\sum_{i=0}^{N-1} I_i \tag{1}$$

$$T = 0.27\mu \tag{2}$$

Pupil is signed as black pixels on image.

In the first case, when the pupil clearly appears on eye image, the results of adaptive threshold itself is able to detect pupil location. Pupil is marked as one black circle on image. By using connected labeling component method, we

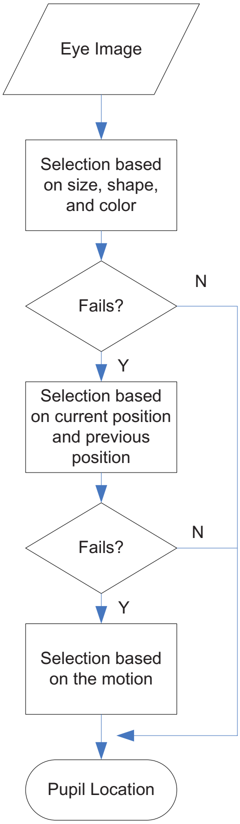

Eye Image
Selection based on size, shape, and color
Fails?
N
Y
Selection based on current position and previous position
Fails?
N
Y
Selection based on the motion
Pupil Location

Fig. 4. Flow of pupil detection. Pupil detection works using three steps: (1) based on size, shape, and color, (2) based on its sequential position, and (3) based on the motion.

can easily estimate the pupil location. While noise appears on image, we can distinguish them by estimate its size and shape. This case is shown in Fig. 5.

Next case, when eye is looking at right or left, the form of eye will changes. This condition makes the pupil detection is hard to find. Noise and interference between pupil and eyelid appear. This condition bring through others black pixels which have same size and shape with pupil. To solve this problem, we utilize the previous pupil location. The pupil is decided using following equation,

$$P(t-1) - C < P(t) < P(t-1) + C \tag{3}$$

The reasonable pupil location P(t) always in surrounding previous location P(t-1) with the area C. This case is shown in Fig. 6.

Last case, when all steps above fail to detect pupil location, we estimate pupil location by its motion. This situation happens when the black pixels mixed with others or no black pixels at all on image. We put this knowledge as last priority

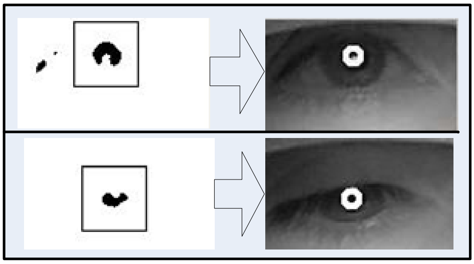


Fig. 5. Case 1. This figure shows clearly pupil and eye is wide open.

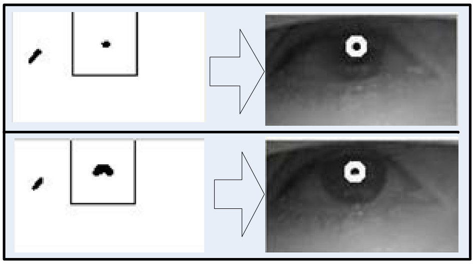


Fig. 6. Case 2. This figure shows pupil and noise have same size and shape.

to avoid ambiguity motion between pupil and other eye components. This case is shown in Fig. 7. This figure shows that no black pixels as pupil on images (left side).

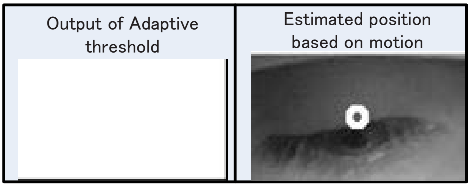

Output of Adaptive threshold
Estimated position based on motion

Fig. 7. Case 3. This figure shows pupil is too small.

We monitor pupil location using its previous location. We adopt Kalman filter [18] [28] to estimate pupil location. The motion of a pupil at each time instance (frame) can be characterized by its position and velocity. Let (i_t, j_t) represent the pupil pixel position (its centroid) at time t and (u_t, v_t) be its velocity at time t in i and j directions. The state vector at time t can therefore be represented as $x_t = (i_t \; j_t \; u_t \; v_t)^t$. The system can therefore be modeled as,

$$x_t + 1 = \mathit{\Phi} x_t + w_t \tag{4}$$

where w_t represents system perturbation. We assume that a fast feature extractor estimates $z_t = (\hat{i_t}, \hat{j_t})$, the pupil position at time t. Therefore, the measurement model in the form needed by the Kalman filter is,

$$z_t = Hx_t + u_t \tag{5}$$

where u_t represents measurement uncertainty. When system is running, we storage pupil location as history. Given the state model in equation 1 and measurement model in equation 2 as well as some initial conditions, the state vector x_t+l, along with its covariance matrix $\Sigma t + 1$, can be updated using the system model (for prediction) and measurement model (for updating).

3.3 Eye Model

A simple eye model is defined on Fig. 8. The eyeball is assumed to be a sphere with radius R. Actually, it is not quite a sphere but this discrepancy does not affect our methodology. The pupil is located at the front of eyeball. The distance from the center gaze to current gaze is r. Gaze is defined as angle θ between normal gaze and r. The relation between R, r and θ is:

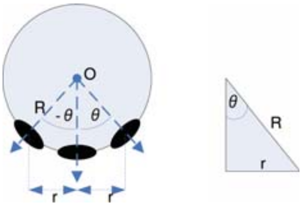


Fig. 8. Eye model

$$r = R \sin \theta \tag{6}$$

$$\theta = \arcsin{(r/R)} \tag{7}$$

The radius of the eyeball ranges from 12 mm to 13 mm according to the anthropometric data [29]. Hence, we use the anatomical average assumed in [30] into our algorithm. Once r has been found, gaze angle θ is calculated easily.

In equation 7 is shown that eye gaze calculated based on r value. In order to measure r, the normal gaze should be defined. In our system, when system start running, the user should looks at the center. At this time we record that this pupil location is normal gaze position. In order to avoid error when acquiring normal gaze, normal gaze position is verified by compare between its value and center of two eye corners.

4 Experiments

The experiments are carried out with five different users who have different race and nationality: Indonesian, Japaneses, Srilanka, and Vietnamese. We collect data from each user while making several eye movement.

Three of Indonesian eye who have different race are collected as shown in Fig. 9. The collected data contain several eye movement such as look at forward, right, left, down, and up. Two of Indonesian have width eye and clear pupil. Number of images are 552 samples and 668 samples. Another Indonesian has slanted eyes and the pupil is not so clear. Number of images of this user are 882 samples. We also collected data from Srilanka people as shown in Fig. 10. His skin color is black with thick eyelid. Number of images are 828 samples. Collected data of Japanese is shown in Fig. 11. His skin color is bright with slanted eyes. Number of images are 665 samples. The last data is Vietnamese as shown in Fig. 12.

The first experiment investigates the pupil detection accuracy and variance against various users. We count the success samples followed by counting the success rate. Our method is compared with adaptive threshold method and

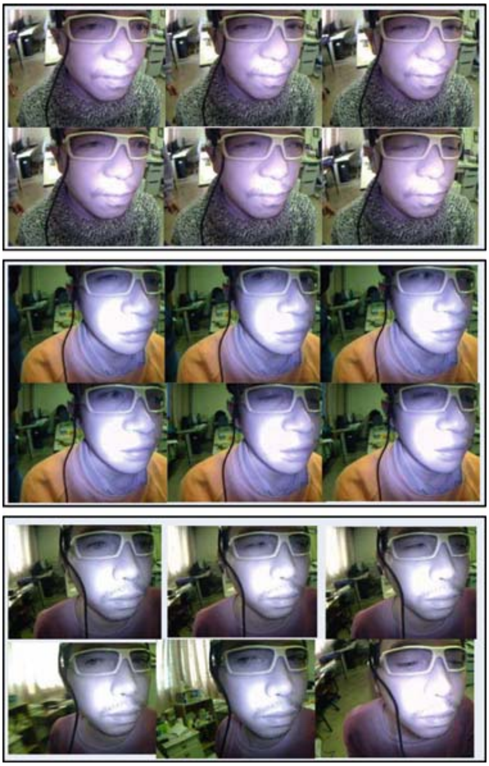


Fig. 9. Example of Indonesian Images

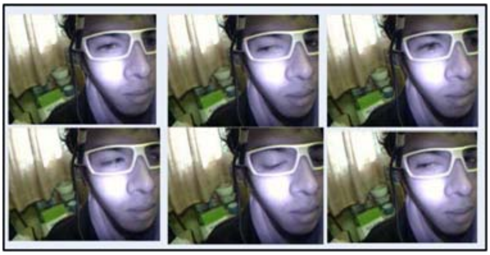


Fig. 10. Example of Srilanka Images

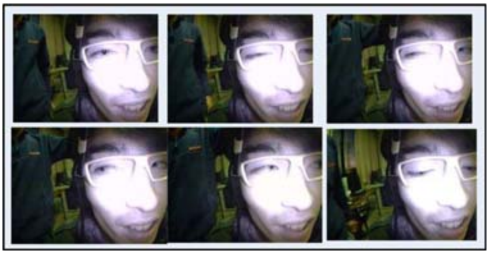


Fig. 11. Example of Japaneses Images

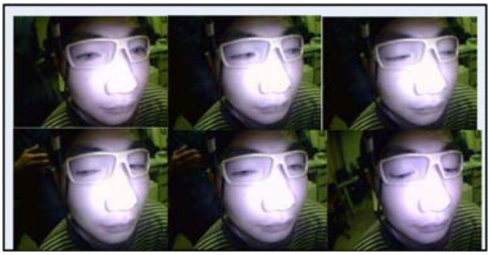


Fig. 12. Example of Vietnamese Images

Template matching method. The adaptive threshold method uses combination between adaptive threshold itself and connected labeling method. The template matching method use pupil template as reference and matched with the images. The result data is shown in table 1.

The result data show that our method has high success rate than others. Also our method is robust against the various users (the variance value is 16.27).

The second experiment measures influence of illumination changes against gaze estimation success rate. This experiment measures the performance of our method when used in different illumination condition. Adjustable light source is given and recorded the degradation of success rate. In order to measure illumination condition, we used Multi-functional environmental detector LM-8000.

Table 1. Robustness against various users. This table shows that our method robust enough against varies user and also has high success rate.

User Types	Nationality	Adaptive Threshold(%)	Template Matching(%)	Our Method(%)
1	Indonesian	99.85	63.04	99.99
2	Indonesian	80.24	76.95	96.41
3	Srilanka	87.80	52.17	96.01
4	Indonesian	96.26	74.49	99.77
5	Japanese	83.49	89.10	89.25
6	Vietnamese	98.77	64.74	98.95
Average		91.07	70.08	96.73
Variance		69.75	165.38	16.27

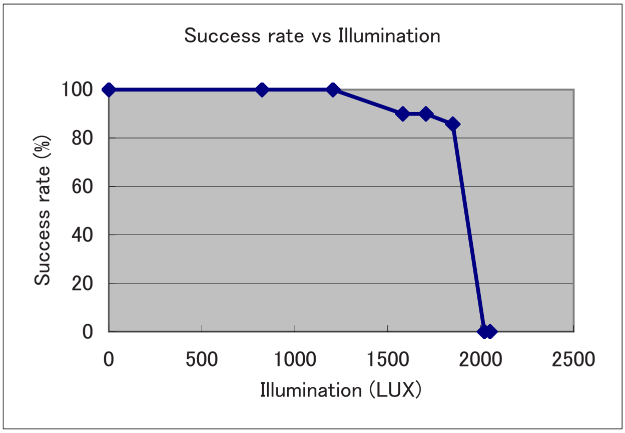


Fig. 13. Illumination Influence. This figure shows that our proposed method works with minimum illumination. The proposed method fails when strong light hit the camera. Condition such this may happens when sun light hit directly into camera.

Experiment data is shown in Fig. 13. Data experiment show that our proposed method works with zero illumination condition (dark place). This ability is caused of IR light source which automatically adjust the illumination. Our proposed method will fails when illumination condition is too strong. This condition may happens when sun light hit directly into camera.

The last experiment measures influence of shock/vibration. This experiment is conducted by giving four types of vibration, and recorded it. Shock recorder G-MEN DR10 was used to measuring the vibration. Experiment data is shown in Fig. 14. Experiment data show that our system is not influenced by vibration. Even vibration/shock happen, user body reduces the vibration and caused the camera sensor stable.

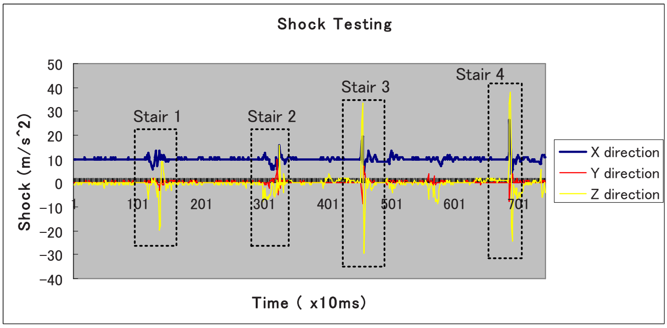


Fig. 14. Shock Influence. This figure shows that our proposed method permit shock or vibration.

5 Discussion

The proposed method has been successfully implemented and shows good performance, robustness, stability, and without any prior calibration process. Our method can improve robustness of existing method against various user problem. Our method perform when other methods fail to detect pupil location. It is caused that our method using three priorities step in order to detect the pupil. Size, shape, and color as first priority perform when user image is clear and less of noise. When the noise and disturbance which come from other eye components appear, our next priority (using its location and motion) success to eliminate them. Also, when adaptive threshold step doesn't yields any black pixels as pupil, our method still performs in order to detect the pupil by using it motion.

6 Conclusion

Eye gaze estimation system using pupil knowledge has been successfully implemented. It shows good performance, is robust to various users, illumination changes, and is robust showing success rate greater than 96% and variance is small enough. Also, calibration process is not required. Furthermore, the proposed system robust against head movement and vibration. Based on these results, we think that our method can be implemented for real-time human computer interaction application. However, this system still need some improvement, such as how to estimate the pupil location when eye is too narrow. Also, when the user has very thick eyelids and make eye is not clear. Next the research will focus on improvement of gaze estimation accuracy. Moreover, we will test the improvement method using other races.

References

1. Park, K.S., Lee, K.T.: Eye controlled human/computer interface using the line of sight and intentional blink. Computer Ind. Eng. 30(3), 463–473 (1996)
2. Ito, Sudo, Ifuku: The look input type communication equipment for serious physically handicapped persons. The Institute of Electronics, Information and Communication Engineers paper magazine J83D1C5, 495–503 (2000)
3. Yamada, F.: The text creation and the peripheral equipment control device by eye movement. The Institute of Electronics, Information and Communication Engineers paper magazine CJ69D(7), 1103–1107 (1986)
4. Yamada, M.: The research trend of the latest eye movement, An Electric Information-and-Telecommunications Academic Journal, MBE95-132, NC95-90,145-152 (1995)
5. Abe, K., Ohiamd, S., Ohyama, M.: An Eye-gaze Input System based on the Limbus Tracking Method by Image Analysis for Seriously Physically Handicapped People. In: 7th ERCIM Workshop User Interface for All Adjunct Proc., pp. 185–186 (2002)
6. Creative Advanced Technologies, http://www.creact.co.jp/jpn/por.pdf
7. Kuno, Y., Fujii, K., Uchikawa: Development of the look input interface using EOG. The Information Processing Society of Japan Paper Magazine C39C5, 1455–1462 (1998)
8. Robinson, D.A.: A method of measuring eye movement using a sclera search coil in a magnetic field. IEEE Trans. on Biomedical Electronics 10, 137–145 (1963)
9. The organization of the retina and visual system, http://webvision.med.utah.edu/
10. Ito, N.: Eye movement measurement by picture taking in and processing via a video capture card. An Institute of Electronics, Information and Communication Engineers Technical Report C102 C128 C31-36 (2002)
11. Kishimoto, Y., Hirose, C.: Development of the look input system by a cursor move system. The Institute of Image Information and Television Engineers C55C6C917-919 (2001)
12. Corno, L., Farinetti, I.,, S.: A Cost-Effective Solution for Eye-Gaze Assistive Technology. In: Proc. IEEE International Conf. on Multimedia and Expo., vol. 2, pp. 433–436 (2002)
13. Abe, O., Oi, D.: The look input system using the sclera reflection method by image analysis. The Institute of Image Information and Television Engineers 57(10), 1354–1360 (2003)
14. Abe, Daisen, Oi.: The multi-index look input system which used the image analysis under available light. The Institute of Image Information and Television Engineers, 58C11C 1656-1664 (2004)
15. Abe, Daisen, Oi.: The look input platform for serious physically handicapped persons, Human Interface Society Human interface symposium 2004 collected papers C1145-1148 (2004)
16. Bradski, G., Kaebler, A.: Learning Computer Vision with the OpenCV Library, pp. 214–219. O'Reilly, Sebastopol (2008)
17. Haro, A., Flickner, M., Essa, I.: Detecting and Tracking Eyes By Using Their Physiological Properties, Dynamics, and Appearance. In: Proceeding of CVPR 2000, pp. 163–168 (2000)
18. Zhu, Z., Ji, Q., Fujimura, K., Lee, K.: Combining Kalman filtering and mean shift for real time eye tracking under active IR illumination. In: Proceeding of 16th Pattern Recognition International Conference, vol. 4, pp. 318–321 (2002)

19. Takegami, T., Gotoh, T., Kagei, S., Minamikawa-Tachino, R.: A Hough Based Eye Direction Detection Algorithm without On-site Calibration. In: Proceeding of 7th Digital Image Computing: Techniques and Applications, pp. 459–468 (2003)
20. Lam, K.M., Yan, H.: Locating and extracting eye in human face images. Pattern Recognition 29(5), 771–779 (1996)
21. Chow, G., Li, X.: Towards a system of automatic facial feature detection. Pattern Recognition (26), 1739–1755 (1993)
22. Rajpathaka, T., Kumarb, R., Schwartzb, E.: Eye Detection Using Morphological and Color Image Processing. In: Proceeding of Florida Conference on Recent Advances in Robotics (2009)
23. Brunelli, R., Poggio, T.: Face Recognition: Features versus templates. IEEE Trans. Patt. Anal. Mach. Intell. 15(10), 1042–1052 (1993)
24. Beymer, D.J.: Face Recognition under varying pose. In: Beymer, D.J. (ed.) IEEE Proceedings of Int. Conference on Computer Vision and Pattern Recognition (CVPR 1994), Seattle, Washington, pp. 756–761 (1994)
25. Shafi, M., Chung, P.W.H.: A Hybrid Method for Eyes Detection in Facial Images. International Journal of Electrical, Computer, and Systems Engineering, 231–236 (2009)
26. Morimoto, C., Koons, D., Amir, A., Flickner, M.: Pupil detection and tracking using multiple light sources. Image and Vision Computing 18(4), 331–335 (2000)
27. Yuille, A., Haallinan, P., Cohen, D.S.: Feature extraction from faces using deformable templates. In: Proceeding of IEEE Computer Vision and Pattern Recognition, pp. 104–109 (1989)
28. http://citeseer.ist.psu.edu/443226.html
29. Kim, K.-N., Ramakrishna, R.S.: Vision-based eye gaze tracking for human computer interface. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, vol. 2, pp. 324–329 (1999)
30. Newman, R., Matsumoto, Y., Rougeaux, S., Zelinsky, A.: Real-time stereo tracking for head pose and gaze estimation. In: Proceedings of Fourth International Conference on Automatic Face and Gesture Recognition, pp. 122–128 (2000)

Behaviour-Based Web Spambot Detection by Utilising Action Time and Action Frequency

Pedram Hayati, Kevin Chai, Vidyasagar Potdar, and Alex Talevski

Anti-Spam Research Lab (ASRL)
Digital Ecosystem and Business Intelligence Institute
Curtin University, Perth, Western Australia
{pedram.hayati,kevin.chai}@postgrad.curtin.edu.au,
{v.potdar,a.talevski}@curtin.edu.au

Abstract. Web spam is an escalating problem that wastes valuable resources, misleads people and can manipulate search engines in achieving undeserved search rankings to promote spam content. Spammers have extensively used Web robots to distribute spam content within Web 2.0 platforms. We referred to these web robots as spambots that are capable of performing human tasks such as registering user accounts as well as browsing and posting content. Conventional content-based and link-based techniques are not effective in detecting and preventing web spambots as their focus is on spam content identification rather than spambot detection. We extend our previous research by proposing two action-based features sets known as *action time* and *action frequency* for spambot detection. We evaluate our new framework against a real dataset containing spambots and human users and achieve an average classification accuracy of 94.70%.

Keywords: Web spambot detection, Web 2.0 spam, spam 2.0, user behaviour.

1 Introduction

Web spam is a growing problem that wastes resources, misleads people and can trick search engines algorithms to gain unfair search result rankings [1]. As a result, spam can decrease the quality and reliability of the content in the World Wide Web (WWW). As new web technologies emerge, new spamming techniques have also emerged to misuse these technologies [2]. For instance, collaborative Web 2.0 websites have been targeted by *Spam 2.0* techniques. Examples of Spam 2.0 techniques would include creating fake and attractive user profiles in social networking websites, posting promotional content in forums and uploading advertisement comments within blogs.

While similar to traditional spam, Spam 2.0 poses some additional problems. Spam content can be added to legitimate websites and therefore influence the quality of content within the website. A website that contains spam content can lose its popularity among visitors as well as being blacklisted for hosting unsolicited content if the website providers are unable to effectively manage spam.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 351–360, 2010.

A Spam 2.0 technique that has been used extensively is *Web Spambots* or *Spam Web Robots*(which we refer to as *spambots*). Web robots are automated agents that can perform a variety of tasks such as link checking, page indexing and performing vulnerability assessment of targets [3]. However spambots are specifically designed and employed to perform malicious tasks i.e. to spread spam content in Web 2.0 platforms [4]. They are able to perform human-users tasks on the web such as registering user accounts, searching/submitting content and to navigate through websites. In order to counter the Spam 2.0 problem from its source, we focus our research efforts on spambot detection.

Many countermeasures have been used to prevent general web robots from the website [5-7]. However, such solutions are not sophisticated enough to deal with evolving spambots and existing literature lacks specific work dedicated to spambot detections within Web 2.0 platforms.

The study performed in this paper continues from our previous work on spambot detection [4] and presents a new method to detect spambot based on web usage navigation behaviour. The main focuses and contributions of this paper are to:

- Propose a behaviour based detection framework to detect spambots on the Web.
- Present two new feature sets that formulate spambot web usage behaviour.
- Evaluate the performance of our proposed framework with real data.

We extract feature sets from web usage data to formulate web usage behaviour of spambots and use Support Vector Machine (SVM) as a classifier to distinguish between human users and spambots. Our result is promising and shows a 94.70% average accuracy in spambot classification.

2 Spambot Detection

As previously discussed, one area of research to counter the Spam 2.0 problem is spambot detection. The main advantage of such an approach is to stop spam at the source so spambots do not continue to waste resources and mislead users. Additionally, spammers have shown that they use variety of techniques to bypass content-based spam filters (e.g. word-salad [8], Naïve Bayas poisoning [9]) hence spambot detection can be effective solution for the current situation.

The aim for spambot detection is to classify spambot user from human users while they are surfing a website. Some practical solutions such as *Completely Automated Public Turing test to tell Computers and Human Apart(CAPTCHA)[5]*, *HashCode[6]*¸ *Noune[10]*, *Form Variation [10]*, *Flood Control [10]* have been proposed to either prevent or slow down spambots activity within a website. Additionally the increasing amounts of Spam 2.0 and recent works prove that such techniques are not effective enough for spambot detection [11].

Behaviour-based spam detection has more capabilities to detect new spamming patterns as well as early detection and adaptation to legitimate and spam behaviour [12]. In this work we propose behaviour-based spambot detection method based on web usage data.

2.1 Problem Definition

We can formulate spambot detection problem in to a binary classification problem similar to the spam classification problem describe in [13]:

$$D = \{u_1, u_2, \ldots, u_{|U|}\} \quad (1)$$

where,

D is a dataset of users visiting a website

u_i is the i^{th} user

$$C = \{c_h, c_s\} \quad (2)$$

where,

C refers overall set of users

c_h refers to human user class

c_s refers to spambot user class

Then the decision function is

$$\phi(u_i, c_j) : D \times C \rightarrow \{0,1\} \quad (3)$$

$\phi(u_i, c_j)$ is a binary classification function, where

$$\phi(u_i, c_j) = \begin{cases} 1 & u_i \in c_s \\ 0 & otherwise \end{cases} \quad (4)$$

In spambot detection each u_i belongs to one and only one class so, the classification function can be simplified as $\phi(u_i)_{spam} : D \rightarrow \{0,1\}$.

3 Behaviour-Based Spambot Detection

3.1 Solution Overview

Our fundamental assumption is that spambots behave differently to human users within Web 2.0 applications. Hence by evaluating web usage data of spambots and human users, we believe we can identify the spambots. Web usage data can be implicitly gathered while users and spambots surf though websites. However,web usage data by itself is not effective in distinguishing spambot and human users. Additional features need to be evaluated with web usage data in Web 2.0 applications for effective spambot detection. Therefore, we investigate two new feature sets called *Action Time* and *Action Frequency* in study spambot behaviour. An *Action* can be defined as a user set of requested web objects in order to perform a certain task or purpose. For instance, a user can navigate to the registration page in an online forum, fill in the required fields and press on submit button in order to register a new user account. This procedure can be formulated as "*Registering a user account*" action.

Actions can be a suitable discriminative feature to model user behaviour within forums but can also be extendible to many other Web 2.0 platforms. For instance, the "*Registering a user account*" action is performed in numerous Web 2.0 platforms, as users often need to create an account in order to read and write content.

In this work we make a use of *action time* and *action frequency* to formulate web usage behaviour. *Action time* is amount of time spend on doing a particular action. For instance, in "*Registering a new user account*" action, *action time* is the amount of time user spends navigating to account registration page, completing the web form and submitting the inputted information. Similarly, *action frequency*¸ is the frequency of doing one certain action. Additionally, if a user registers two accounts, their "*Registering a new user account*" *action frequency* is two. Section 3.2 provides a formal explanation of *action time* and *action frequency*.

It is possible to classify spambots from human user once *action time* and *action frequency* are extracted from web usage data and feed into the SVM classifier.

3.2 Framework

Our proposed framework consists of 4 main modules, which include *web usage tracking*, *data preparation*, *feature measurement* and *classification*as shown in Figure 1.

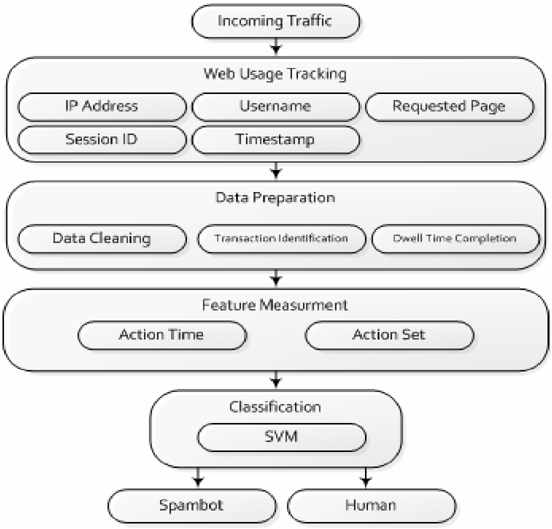


Fig. 1. Behaviour-Based Spambot detection framework

Incoming Traffic

Incoming traffic shows a user entering the website through a web interface such as the homepage of an online forum.

Web Usage Tracking

This module records web usage data including the user's *IP address*, *username*, *requested webpage URL*, *session identity*, and *timestamp*. The username and session ID makes it possible to track each user when he/she visits the system.

Conventionally, web usage navigation tracking is done through web server logs[14]. However, these logs can not specify usernames and sessions of each request. Hence we employ our own web usage tracking system developed in our previous work, *HoneySpam 2.0*[4] in order to collect web usage data.

Data Preparation

This module includes three components, which are *data cleaning*, *transaction identification* and *dwell time completion*.

Data cleaning

This component removes irrelevant web usage data from:

- Researchers who monitor the forum
- Visitors who did not create a user account
- Crawlers and other Web robots that are not spambots

Transaction Identification

This component performs tasks needed to make meaningful clusters of user navigation data[15]. We group web usage data into three levels of abstraction, which include *IP*, *User* and *Session*. The highest level of abstraction is IP and each IP address in web usage data consist multiple users. The middle level is the user level and each user can have multiple browsing sessions. Finally, the lowest level is the session level which contains detailed information of how the user behaved for each website visit.

In our proposed solution we performed spambot detection at the session level for the following reasons:

- The session level can be built and analysed quickly while other levels need more tracking time to get a complete view of user behaviour.
- The session level provides more in-depth information about user behaviour when compared with the other levels of abstraction.

Hence we define a transaction as a set of webpages that a user requests in each browsing session and extract features accordingly.

Dwell time completion

Dwell time is defined as an amount of time user spend on specific webpage in his/her navigation sequence. It can be calculated by looking at each record timestamp. Dwell time is defined as:

$$d'_i = t_{i+1} - t_i \;\; where\, 1 \le i \le |S| \tag{5}$$

d'_i is dwell time for ith requested webpage in session S at time t;

In E.q.5. it is not possible to calculate dwell time for the last visited page in a session. For example, a user navigates to the last page on the website then closes his/her web browser. Hence we consider the average dwell time spent on other webpages in the same session as the dwell time for the last page.

Feature Measurement

This module extracts and measures our proposed feature sets of *action time* and *action frequency* from web usage data.

Definition 1: Set of actions (s_i)

Given a set of webpages $W = \{w_1, w_2, ..., w_{|W|}\}$, A is defined as a set of *Actions*, such that

$$A = \{a_i | \, a_i \subset W\} = \{\{w_l, ..., w_k\}\} \quad 1 \le l, k \le |W| \tag{4}$$

Respectively s_i is defined as

$$s_i = \{a_j\} \quad 1 \le i \le |T|; \quad 1 \le j \le |A| \tag{5}$$

s_i refers to a set of actions performed in transaction i and T is total number of transactions.

Definition 2: Action Frequency ($\overrightarrow{aF} = \left\langle h_1^i, ..., h_{|A|}^i \right\rangle$)

Action frequency ($\overrightarrow{aF}$) is a vector where h_j^i is the frequency of j^{th} action in s_i. otherwise it is zero.

Definition 3: Action Time ($\overrightarrow{aT} = \left\langle d_1^i, ..., d_{|A|}^i \right\rangle$)

We define action time as a vector where

$$d_j^i = \begin{cases} \dfrac{\sum_{k \in a_j} d'_k}{h_j^i} & a_j \in s_i \\ 0 \quad otherwise \end{cases} \tag{7}$$

d_j^i is a dwell time for action a_j in s_i which is equal to total amount of time spend on each webpage inside a_j. In cases that a_j occurs more than once, we divide d_j^i by the action frequency, h_j^i to calculate the average dwell time.

Classification

We employ Support Vector Machine (SVM) as our machine learning classifier. Support Vector Machine (SVM) is a machine learning algorithm designed to be robust for classification especially binary classification [16]. SVM trains by n data points or features $\{(x_1, y_1), (x_2, y_2),...(x_n, y_n))$ and each feature comes along with class label (y_i). As mentioned in previous section there are two classes {*human,spambot}* in spambot detection, which we assign numerical value -1 and +1 to each class accordingly. SVM then tries to find an optimum hyperplane to separate two classes and maximising the margin between each class. A decision function on new data point x is define as $\phi(x) = \operatorname{sgn}(w, x + b)$ where *w* is weight vector and *b* is bias term.

3.3 Performance Measurement

We utilised *F-Score* to measure the performance of our classification results [17]. F-Score is defined E.q. 8.

$$F = 2\frac{\mathrm{Re}\,call \times \mathrm{Pr}\,ecision}{\mathrm{Re}\,call + \mathrm{Pr}\,esision} \tag{8}$$

where

$$\mathrm{Re}\,call = \frac{|Detected\ Spambots|}{|Spambots|}$$

$$\mathrm{Pr}\,ecision = \frac{|Detected\ Spambots|}{|Spambots + Humans|}$$

In the next section we discuss about experimental result of our work.

4 Experimental Results

4.1 Data Set

We collected our spambot data from our previous work [4] over a period of a month. We combine this data with human data collected from an online forum with same configuration as spambot host. We removed domain specific information from both datasets. Next we combined these two dataset for experimentation. Table 1 illustrates a summary of our collected data.

Table 1. Summary of collected data

Data	Frequency
# of human records	5555
# of spambot records	11039
# of total sessions	4227
# of actions	34

In feature measurement module we come up with 34 individual actions. We extract *action time* and *action frequency* from our dataset and use them separately in our classifier.

4.2 Results

We run 2 experiments on our dataset based on each feature set. We achieved an average accuracy of 94.70%, which ranges from 93.18% for *action time* to 96.23% for *action frequency*. Table 2 and Table 3 summarise the result from each experiment along with ratio of true-positives(TP) (the number of correctly classified spambots) and false-positives (FP) (number of incorrectly classified human users).

Table 2. Summary of experimental results on *action time* (aT) feature set

C	TP	FP	Precision	Recall	*F*
c_h	0.976	0.399	0.948	0.976	0.962
c_s	0.601	0.024	0.774	0.601	0.676
Average	0.932	0.355	0.927	0.932	0.928

Table 3. Summary of experimental results on *action frequency* (aF) feature set

C	TP	FP	Precision	Recall	*F*
c_h	0.998	0.299	0.961	0.998	0.979
c_s	0.701	0.002	0.975	0.701	0.815
Average	0.962	0.264	0.963	0.962	0.960

It is clear that *action frequency* is a slightly better classification feature to classify spambot from human users. Spambots tend to repeat certain tasks more often when compared with humans that perform a larger variety of tasks rather than focusing on specific tasks. The result of our work shows that *action time* and *action frequency* are good feature for spambot detection and therefore Spam 2.0 prevention.

5 Related Works

There has been extensive research focused on spam management and spam filtering. However, there has been little work dedicated to Spam 2.0 and spambot detection.

In the web robot detection, Tan et al. [3] propose a framework to detect unseen and camouflaged web robots. They use navigation pattern, session length and width as well as the depth of webpage coverage to detect web robots. Park et al. [7] present a malicious web robot detection method based on HTTP headers and mouse movement. However none of these works have studied spambots in Web 2.0 applications.

Yiqun et al.[18] and Yu et al. [19] utilise user web access logs to classify web spam from legitimate webpages. However the focus of their work is different from ours as they rely on user web access log as a trusted source for web spam classification. However, in this work we show that web usage logs can be obtained from both humans and spambots and such as distinction should be made.

In our previous work on HoneySpam 2.0 [4], we propose a web tracking system to track spambot data. The dataset collected in HoneySpam 2.0 is used in this work.

6 Conclusion

To the best of our knowledge, our research from [4] and this paper is the first work focused on spambot detection while conventional research has been focused on spam content detection. In this paper, we extended our previous work in spambot detection in Web 2.0 platforms by evaluating two new feature sets known as action time and action frequency. These feature sets offer a new perspective in examining web usage data collected from both spambots and human users. Our proposed framework was validated against an online forum and achieved an average accuracy of 94.70% and evaluated the performance of our framework using F-score. Future work will be focused on evaluating more feature set or combination of feature set, decrease ratio of false-positives as well as extending our work on other web 2.0 platforms to classify spambots from human users.

References

[1] Gyongyi, Z., Garcia-Molina, H.: Web spam taxonomy. In: Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web, Chiba, Japan (2005)

[2] Hayati, P., Potdar, V.: Toward Spam 2.0: An Evaluation of Web 2.0 Anti-Spam Methods. In: 7th IEEE International Conference on Industrial Informatics Cardiff, Wales (2009)

[3] Tan, P.-N., Kumar, V.: Discovery of Web Robot Sessions Based on their Navigational Patterns. Data Mining and Knowledge Discovery 6, 9–35 (2002)

[4] Hayati, P., Chai, K., Potdar, V., Talevski, A.: HoneySpam 2.0: Profiling Web Spambot Behaviour. In: 12th International Conference on Principles of Practise in Multi-Agent Systems, Nagoya, Japan, pp. 335–344 (2009)

[5] von Ahn, L., Blum, M., Hopper, N., Langford, J.: CAPTCHA: Using Hard AI Problems for Security. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 646–646. Springer, Heidelberg (2003)

[6] Mertz, D.: Charming Python: Beat spam using hashcash, (2004), (August 3, 2009) http://www.ibm.com/developerworks/linux/library/l-hashcash.html (Accessed)

[7] Park, K., Pai, V.S., Lee, K.-W., Calo, S.: Securing Web Service by Automatic Robot Detection. In: USENIX 2006 Annual Technical Conference Refereed Paper (2006)

[8] Uemura, T., Ikeda, D., Arimura, H.: Unsupervised Spam Detection by Document Complexity Estimation. In: Discovery Science, pp. 319–331 (2008)

[9] Sarafijanovic, S., Le Boudec, J.-Y.: Artificial Immune System for Collaborative Spam Filtering. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2007), pp. 39–51 (2008)

[10] Ogbuji, U.: Real Web 2.0: Battling Web spam (August 3, 2009) (2008), http://www.ibm.com/developerworks/web/library/wa-realweb10/ (Accessed)

[11] Abram, H., Michael, W.G., Richard, C.H.: Reverse Engineering CAPTCHAs. In: Proceedings of the 2008 15th Working Conference on Reverse Engineering, vol. 00. IEEE Computer Society, Los Alamitos (2008)

[12] Salvatore, J.S., Shlomo, H., Chia-Wei, H., Wei-Jen, L., Olivier, N., Ke, W.: Behavior-based modeling and its application to Email analysis. ACM Trans. Internet Technol. 6, 187–221 (2006)

[13] Le, Z., Jingbo, Z., Tianshun, Y.: An evaluation of statistical spam filtering techniques. ACM Transactions on Asian Language Information Processing (TALIP) 3, 243–269 (2004)

[14] Cooley, R., Mobasher, B., Srivastava, J.: Data Preparation for Mining World Wide Web Browsing Patterns. Knowledge and Information Systems 1, 5–32 (1999)

[15] Cooley, R., Mobasher, B., Srivastava, J.: Web mining: information and pattern discovery on the World Wide Web. In: Proceedings of Ninth IEEE International Conference on Tools with Artificial Intelligence 1997, pp. 558–567 (1997)

[16] Chang, C., Lin, C.: LIBSVM: a library for support vector machines, S. a. a. Ed (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm

[17] Rijsbergen, C.J.V.: Information retrieval Butterworths (1979)

[18] Yiqun, L., Rongwei, C., Min, Z., Shaoping, M., Liyun, R.: Identifying web spam with user behavior analysis. In: Proceedings of the 4th international workshop on Adversarial information retrieval on the web, ACM (2008)

[19] Yu, H., Liu, Y., Zhang, M., Ru, L., Ma, S.: Web Spam Identification with User Browsing Graph. Information Retrieval Technology, 38–49 (2009)

Towards the Definition of a Metamodel for the Conceptual Specification of Web Applications Based on Social Networks

Constanza Hernández, Ricardo Quintero, and Leopoldo Z. Sánchez

Department of Information Systems and Computation
Instituto Tecnológico de Culiacán,
Culiacán, Sinaloa, México
chg30@hotmail.com, iscrquinter@prodigy.net.mx,
leopoldo@correo.ccs.net.mx

Abstract. The present work is done within the framework of the Model Driven Development of Software. It proposes an initial strategy for obtaining a metamodel that captures the main elements (objects, actors, activities, subjects, relations, etc.) that characterize the Web applications of Social Networks. It also includes the definition of a tool that allows the graphical edition of models for the mentioned applications, considering as the base for capturing the requirements of the main elements of the Social Network application. With these models, a general automatic code generation strategy for a Web 2.0 application is presented.

Keywords: Modeling, UML, Web 2.0, Social Network, Web-engineering.

1 Introduction

Nowadays the Web 2.0 applications have become very important and popular. The Web has become a platform where the users can contribute with knowledge by means of the publication of contents. A type of Web 2.0 applications very important that have had a great impact are the Online Social Networks, sites of social interaction formed by profiles of users, who encourage friendly groups to share diverse contents and to execute actions over different items: photographs, videos, music, sent messages by means of an internal network of mail in the site and additional functionality that can be integrated in a Social Network application.

Several sites of Social Networks exist, the most popular according to Alexa.com[14] are: Facebook[5], Youtube [8], MySpace [12], Flickr[4][6], among others. The great popularity that these applications have had provides us the opportunity to study its characteristics and properties in order to promote new methodologies or models to carry out the analysis, design and implementation of these applications.

In this paper we show the advances on the area that we are doing. We believe that the base for the correct implementation of social network applications should start in the requirements phase. Our proposal is that if we can model the social community life first (in this phase) then we can be able to generate the Web application that

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 361–369, 2010.

supports the community. Using the MDA approach is possible to define these models for the specification of the social community and then using model transformation mechanism, the Web application could be obtained.

The main contributions of the work are: (1) An initial proposal of social network metamodel and (2) A code generation strategy for the automatic production of the application that support the social community life.

The work is organized in the following way: in section 2 we expose a description of the research work and its methodology; section 3 shows the advances obtained and examples; section 4 shows the main strategy for the code generation and finally, section 5 presents future works and conclusions.

2 Description of the Work and the Methodology

2.1 The Work

One of the purposes of this research is to create a common and integrated set of models that capture all the elements that characterize a social network community giving a complete and coherent representation of this.

From this set of models we have the basis for the definition of the requirements used to carry out the implementation of a tool (Editor and a Model Compiler), another one of the goals of the research.

2.2 The Methodology

One part of the methodology for the definition of the metamodel requires the identification of the elements that characterize the social networks, as well as the possible relations among them. In order to carry out the construction of the metamodel we use the Borland Together 2007 tool based on Eclipse [9] and we used UML [10] as the modeling language. In the case of the tool, this will fulfill the specifications of the Model Driven Architecture (MDA)[7]. The code will be written in the Java language [13] as a plug-in for the Eclipse platform [11].

3 The Research

3.1 State of the Art: Social Network Elements

The first task we have to done is the identification of the Social Network elements. This is because they are the basic elements for the metamodel. There are many contributions that define the elements that these virtual communities have, and some of these agree in certain aspects, others handle different aspects.

Next is a classification of the basic elements that characterize a Social Network base on [1]:

- ***People:*** they are the actors of social interactions. Each actor perform tasks and play roles in the community.
- ***Shared purpose:*** the focus of social interactions; it could be a matter of interest, a job or a service.

- ***Policies:*** the rituals, protocols, laws. etc. that govern people interaction.
- ***Social Software:*** the infrastructure that mediates communication to let people share tasks and “feel together". Community-support applications revolve around a common set of concepts independent of the community's purpose.
- ***Members:*** act, communicate and set relationships, changing the state of the application by their behavior. By acting within the community each member builds up her/his own reputation.
- ***Items:*** generic elements of interest. A community may base its existence on the sharing of items (e.g., bookmarks, videos, pictures, comments).
- ***Activities:*** an action relevant for the development of the community. Some activities could have a positive impact: e.g., item contributions, invitations, tagging, and rating.
- ***Messages:*** Unlike items, messages do not add value to the community directly, but support relationship development.

Also, according to Misole et al [2], other elements typical included in social networks are:

- ***Users:*** who for participating in a social network must register itself in the site, possibly under a pseudonym. Some sites allow navigate in public information without the necessity to register itself. The users can provide voluntary information about them, which is added to a profile of the user.
- ***Links:*** some sites allow the users to connect with some other users, the connections are used to have contact between them, or to share interests in contributed content.
- ***Groups:*** most of the sites of social networks allow their users to create and join to special interest groups; the users can send messages and load information to a group.

Another contribution that we have take into account is the work of Porter Joshua [3] that has served to us to identify elements for our meta-model. Porter propose the AOF method for design an application Social Web [3], which consists of the following three steps:

1. The first step focuses in identifying a primary activity in which one is going away to focus the Web site, although in these sites usually they are realized a great number of activities, is necessary to concentrate in a main activity, like for example the Flickr.com site that concentrates in the sharing of photographs; Youtube.com in videos and thus in another case. These sites have additional activities than the primary activity.
2. In the Second step it is necessary to identify all the social objects that are going to comprise the site or with that the users are going to interact. Between these it is possible to be mentioned: photographs, videos, films, products, messages, events, etc.
3. As third step a set of characteristics have to be selected, these would come to form the actions that will be realized with these social objects that they are extremely important to support in the application. Like an example with an object photo, the actions that can be realized, would be, to send, to unload, to share, etc.

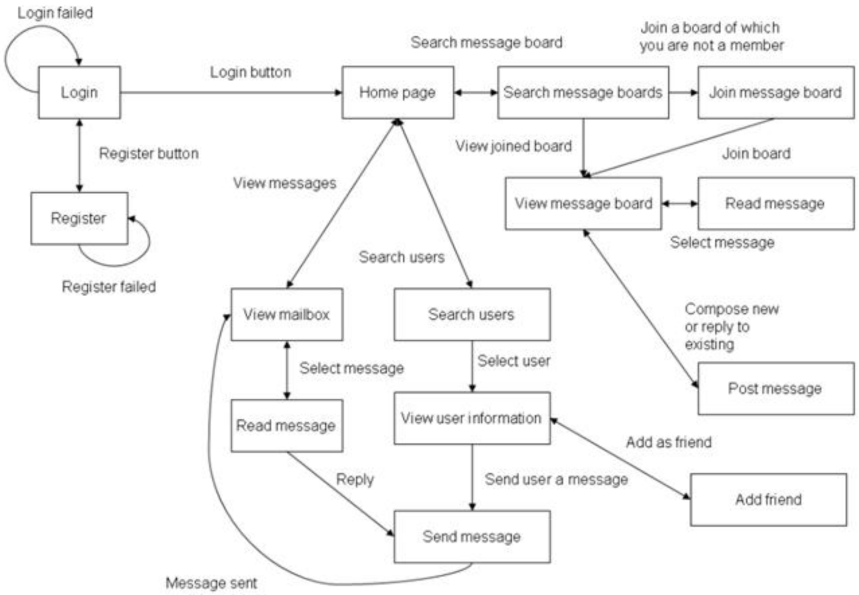


Fig. 1. Basic activities in a social web application

Another aspect that we explored was the dynamical behavior of the social applications. In this case we focus on the typical activities of the users of social Web applications. Figure 1 shows the basic activities for the navigation. The model is similar to a diagram state. It comes from the work of [16].

The initial activity is the registration of the user. This process enables the user to be a member of the social network. Once the user has registered, the user can login in the system using the username and password assigned during the register.

Once in the first page the interface offers to the user the different activities that he can do in the social network: read or post messages, search friends, establish relationships, qualify other participants, etc.

Another strategy that we used identify the elements of the definition of our meta-model was the exploration of sites that involve Social Networks.

As an example Figure 2 shows a basic model built with the tool. The model was constructed on the basis of the work of Uyeda [4].

This example represents the underlying Social Network in the Flickr.com site, which allows its users to organize photos in order to tag them and share. In addition, the photos can be organized in sets, where a set can contain many photos and a photo can be contained in many groups. In addition the users can join to these groups. A group can contain a pool, a collection of photos contributed by the members of that group. In summary, the elements located in this application can be useful to us to carry out the meta-model are: users, groups, photos, set, collection of photos (pool), labels and commentaries, as well as the relationships that occur among them.

As a part of the contribution of this paper we can present a first approach of the metamodel. The metamodel is built using the UML modeling language.

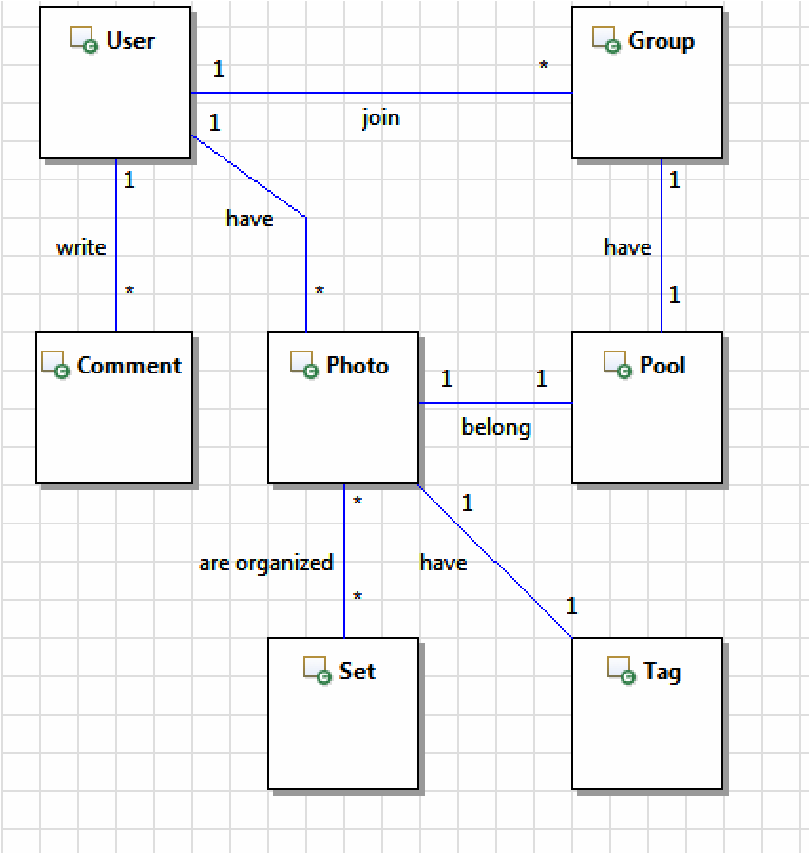


Fig. 2. Modeling example of Flickr.com

Figure 3 shows the necessary elements represented in a social network application. The first element of the metamodel represents the Social Network (`SocialNetwork`). Each Social Network contains many Members. The Members are the Persons that have common interest and register in the social network. The Members interact with social objects (`SocialObject`). These social objects can be a video, articles, book, message, music, event. Each one of the SocialObject is handled with one or many Actions, like load, download, tag, comment, edit, send and so on. Each action is represented by the enumeration class `actionType`. Each Member has a Profile formed with different information like basic, personal, school and job information. This information is represented with the `typeInformation` metaclass.

The `NetworkObject` represents the different services needed to support the activities of the members in the network. These services are typically implemented by group of members, boards, mailbox, comment and so on. Each one of these Network Objects are implemented with one or many Activities For example `createGroup`, `sendMessage`, `handleBoard`, `searchFriends`, `accountRegistry`, etc.

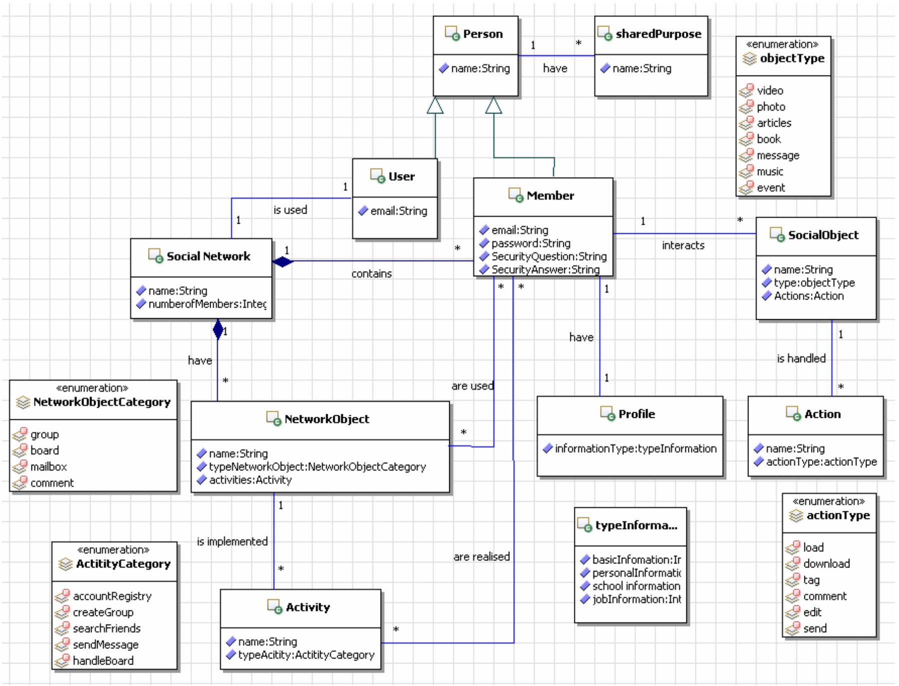


Fig. 3. Initial proposal of Social Network metamodel

The Social Network is used by one or many persons who are users that do not need to register in the Social Network as a Member. These are represented by the `User` metaclass.

We have presented the elements and relationships of the metamodel for modeling Social Network applications. From this metamodel the designer can specify the social network and by using a model compiler generate the Web application that supports the community. This is the next task of our research.

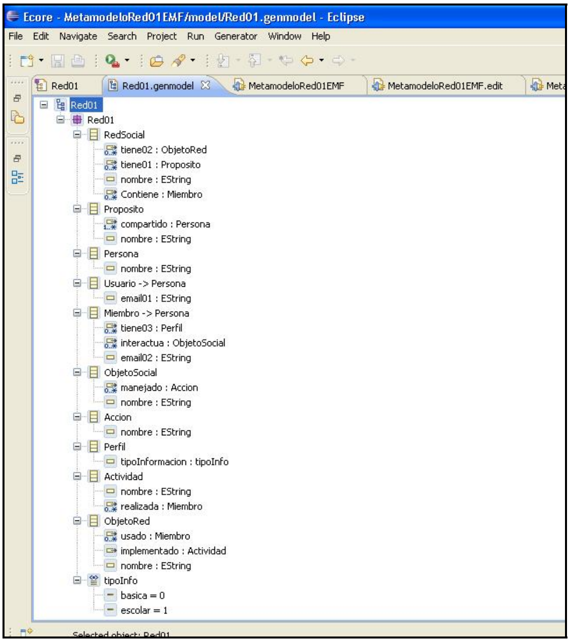
4 Code Generation Strategy

In the MDA context our meta-model would be an example of a Platform Independent Model (PIM). This model represents the Social Network in a language with a high level of abstraction, without making reference to technological platforms. MDA indicates that from this model should be established a set of model transformation rules in order to generate the code for the underlying software application. This will be the basis for code generation strategy of our tool. From the PIM model, a model of lower level of abstraction (or Platform Specific Model - PSM) is obtained by a set of Model to Model transformations and from the PSM the final code will be generated by a set of Model to Text transformations.

The following steps define the basic design of the MDA tool for the implementation of our code generation proposal:

1. **Build a model editor:** using the EMF Eclipse technologies [20] a model editor should be built in which the designer defines the social network model (the PIM). Eclipse offers the Graphical Modeling Framework (GMF) [17] to implement this step.
2. **Define the Model to Model transformations:** Using the QVT [18] language the PIM model is transformed to the PSM model of the different technologies needed to implement the Web Application.
3. **Define the Model to Text transformations:** The code is generated using the Mofscript [19] language. This step is helped with the designing of a framework that reduce the gap between the models and the execution platform.

The construction of the model editor has taken two phases. In the first one we try the native EMF Eclipse tools to built our metamodels. We implement the metamodel using EMF-ecore. Figure 4 shows an example of the editor generated with these tools.



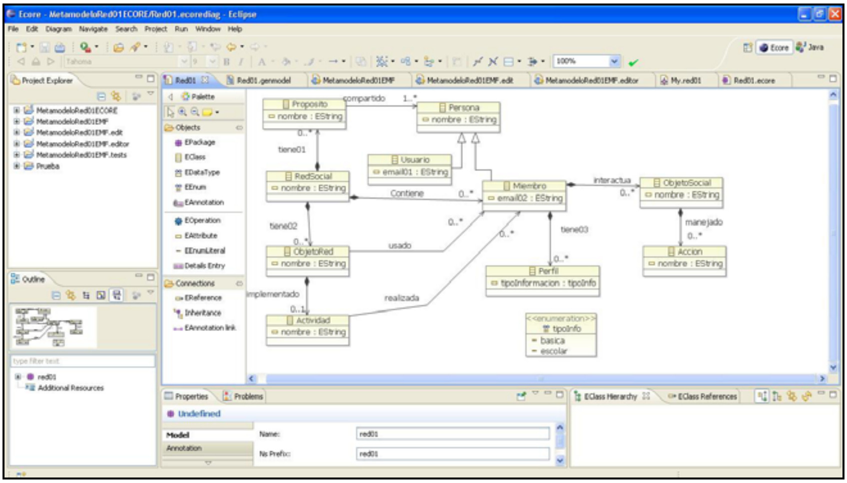
```
Red01
  Red01
    RedSocial
      tiene02 : ObjetoRed
      tiene01 : Proposito
      nombre : EString
      Contiene : Miembro
    Proposito
      compartido : Persona
      nombre : EString
    Persona
      nombre : EString
    Usuario -> Persona
      email01 : EString
    Miembro -> Persona
      tiene03 : Perfil
      interactua : ObjetoSocial
      email02 : EString
    ObjetoSocial
      manejado : Accion
      nombre : EString
    Accion
      nombre : EString
    Perfil
      tipoInformacion : tipoInfo
    Actividad
      nombre : EString
      realizada : Miembro
    ObjetoRed
      usado : Miembro
      implementado : Actividad
      nombre : EString
    tipoInfo
      basica = 0
      escolar = 1
```

Selected object: Red01

Fig. 4. The native EMF editor

The drawback of this kind of editor is that is error prone and difficult to edit each one of the metamodels elements, so we try to design the metamodel using a visual tool.

Figure 5 shows the metamodel editor using a visual approach. Similar to a class diagram, the edition is more direct and easy. The interface also offers different views (windows) to edit the properties of the metamodel elements.



Name: red01

Ns Prefix: red01

Fig. 5. The visual editor of the metamodel

5 Conclusions and Future Work

Considering the growth and importance that the Web 2.0 applications are taking today and the changing dynamics or the same, greater efforts are required in software engineering for their specification, development, construction and maintenance. Considering that there is constant pressure to keep up to date such systems, software engineering requires new methods, techniques and tools for their support. This paper aims to contribute to it.

In this article we have shown the research work we have done at date. As a future work, in regard to the proposed metamodel, we are going to extend it and provide validation supported by the addition of OCL constraints [15]. With respect the tool, after obtaining the metamodel, we are going to hold its full implementation. In this moment of our research we only have implemented the first version of the editor (the first step of our code generation strategy). For the other steps we need to define the PIM to PSM and PSM to Code model transformation rules according to the MDA framework.

Another important task that we need to resolve is the construction of the framework that facilitates the code generation step, the most difficult task for obtaining the final Web application.

References

1. Piero, F., Massimo, T., Matteo, S., Lorenzo, F.: Building Community-Based Web Applications with a Model-Driven Approach and Design Patterns, Politecnico di Milano (2009)
2. Alan, M., Massimiliano, M., Gummadi, K.P.: Measurement and Analysis of Online social Network. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, New York, USA, pp. 29–42 (2008)

3. Joshua, P.: Designing for the social web, pp. 23–40. New Riders (2008)
4. Uyeda, F., Gupta, D., Vahdat, A., et al.: GrassRoots: socially-driven web sites for the masses. In: Proceedings of the 2nd ACM workshop on Online social networks, Barcelona, Spain (2009)
5. The Facebook, Privacy Policy (Agosto 2005), `http://facebook.com/policy.php`
6. Graham, J.: Flickr of idea on a gaming project led to photo website, USA (Febrero 27 Today, (2006)
7. Dragan, G., Dragan, D., Vlandan, D.: Model Driven Architecture and Ontology Development. Springer, Heidelberg (2006)
8. Youtube (Febrero 2005), `http://www.youtube.com/t/about`
9. Borland Software Corporation, Visual modeling platform for software teams, `http://www.spectrum-systems.com/vendors/Borland/together_datasheet_2006.pdf`
10. Fowler, M.: UML distilled Third Edition A brief guide to the standard Object Modeling Language. Addison-Wesley, Reading (2004)
11. Zhixiong, C., Delia, M.: Experiences with Eclipse IDE in programming courses. J. Comput. Small Coll. 21(2), 104–112 (2005)
12. Brian Chatfiel, T.: The MySpace.com Handbook: The Complete Guide for members and parents. Atlantic Publishing Group Inc (2007)
13. Deitel, H.M., Deitel, P.J.: Como programar en Java. Quinta edición. Prentice Hall, Englewood Cliffs (2004)
14. Alexa. Consultado en noviembre de (2009), `http://www.alexa.com`
15. Warmer, J., Clark, T.: Object Modeling with the OCL. LNCS, vol. 2263, Springer, Heidelberg (2002)
16. Design Specifications for Social Networking System, `http://www.cs.ucr.edu/~yuc/dimension/DOC/design_spec/design_spec.pdf`
17. Eclipse Graphical Modeling Framework, `http://www.eclipse.org/modeling/gmf`
18. MOF/QVT – Object Modeling Group, `http://www.omg.org/mda`
19. MOFScript - Eclipse, `http://www.eclipse.org/gmt/mofscript`

Semi-automatic Information Extraction from Discussion Boards with Applications for Anti-Spam Technology

Saeed Sarencheh[2], Vidyasagar Potdar[1], Elham Afsari Yeganeh[2], and Nazanin Firoozeh[2]

[1] Anti-Spam Research Lab (ASRL) Digital Ecosystems and Business Intelligence Institute, Curtin University, Perth, Australia
http://asrl.debii.curtin.edu.au
[2] Institute for Advanced Studies in Basic Sciences, IASBS, Zanjan, Iran
http://ww.iasbs.ac.ir

Abstract. Forums (or discussion boards) represent a huge information collection structured under different boards, threads and posts. The actual information entity of a forum is a post, which has the information about authors, date and time of post, actual content etc. This information is significant for a number of applications like gathering market intelligence, analyzing customer perceptions etc. However automatically extracting this information from a forum is an extremely challenging task. There are several customized parsers designed for extracting information from a particular forum platform with a specific template (e.g. SMF or phpBB), however the problem with this approach is that these parsers are dependent upon the forum platform and the template used, which makes it unrealistic to use in practical situations. Hence, in this paper we propose a semi-automatic rule based solution for extracting forum post information and inserting the extracted information to a database for the purpose of analysis. The key challenge with this solution is identifying extraction rules, which are normally forum platform and forum template specific. As a result we analyzed 72 forums to derive these rules and test the performance of the algorithm. The results indicate that we were able to extract all the required information from SMF and phpBB forum platforms, which represent the majority of forums on the web.

Keywords: Information extraction, Discussion Forums, Anti-Spam, phpBB, SMF.

1 Introduction

Forum is a place that members with different interests can interact in order to participate in discussions, share information and find out important issues. Since forum is used in education, health care, community, business contexts and so on [1], planning a powerful forum is very important. For example http://discussion.forum.nokia.com/forum/ is a commercial forum where people can share their experiences about different series of Nokia platforms. Other one (http://opcwacademicforum.org/) is an academic forum that introduces workshops and has a schedule about tasks done in workshop's days.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 370–382, 2010.

Like these there are millions of other forums on the web. We estimate that there are at least 97 million forums on the Web as of November 2009[1].

These forums are hosted on a number of different platforms or software, the famous ones are SMF, phpBB and vBulletin. In Australian domains these are used more often than any other open source software's. All the forums are predominantly programmed in PHP and use MySQL as a database. phpBB is currently supported by six core developers, more than 40 team members, and 250,000 registered users [2]. SMF also is an open source package which is highly extensible and flexible. It is currently supported by several teams, including a customization team, development team, documentation team, support team, marketing team, project management team and an internationalization team. Compared to phpBB and SMF, vBulletin is a proprietary platform allows forum's owners improve their service to the forum's users. These three forum platforms represent the most popular forum platforms used on the web these days [3].

As it is seen, all these forums contain a wide range of information that is very important and helpful for users. Since forums contain valuable opinions from different people on different topics from different fields, forums play an important role to represent rich source of information. Therefore spammers make use of this valuable tool to achieve their goals. In recent years, spammers have exploited legitimate website security vulnerability to conceal themselves from anti-spam filters and genuine users, for example by injecting junk content into the web site [4]. Immoderate growth of the junk content is degrading the quality of the information is available in web applications, for example spammers can artificially prompt products and services via forums [20, 21]. There is no specific technique for detection and prevention spam in forums [8], for this reason, forum information extraction is the first step towards detecting and preventing forum spam.

This information can be of great significance for many businesses but unfortunately, this information is unstructured and distributed. If this information could be extracted, integrated, mined and presented in a right format, it can be an extremely valuable resource. Business intelligence information can be gathered from this evaluation. It is often seen these days that many companies and organization are trying to integrate backed information to create value e.g. meta search engines, comparative shopping services, gathering market intelligence, conducting customer behavior analysis and studying product bugs. For example, Yahoo and Google use the information that posted in forums for improving the quality of search results especially for Q&A queries [18].

Information Extraction (IE) is an area of information technology that uses machine learning techniques to autonomously extract useful information from the Web. IE is used to select desired fields from the given data, by extracting common patterns that appear along with the information. Examples of IE usage are

⇒ **zoominfo.com:** aggregates information on people in order to allow users enter their target name and shows them title and company which are related to that name,

⇒ **flipdog.com:** gathers data on jobs and categorizes them based on educational jobs, government jobs, etc; it also shows the jobs related to special city or state. Hence by using this site, users can easily find their proper local job.

[1] The number of forums were obtained by adding the number of Google search results returned with the URL pattern of "http://forum.*", "http://forums/*", "*/forum/*", "*/forums/*".

⇒ **CiteSeer.org:** This site extracts number of citation for each academicals research paper. This site read the paper reference and increases the paper's citation number for each reference to that paper. It also does duplicates citation entries from papers' reference sections, so you can easily find all the papers that cite a certain paper. So, a graph of citation for a paper can be drawn and fined seminal paper in a subfield, so can listed a related paper for user. Other similar services include rexa.info[5], [6].

It is also used in areas such as sale products indexing, job advertisement collection and scientific article collection from the Internet, among several others [7]. Automatically "fill-in" database forms from unstructured data such as Web documents or email is another usage of IE [19]. Since web mining systems retrieve metadata from textual information in Web pages by using IE techniques, it is used in semantic web too [7].

So, we want to use rule based algorithms to extract information from forums. There are two topics that lead us to make a rule based algorithms, *firstly* the style of forums are changed rarely and all of the template should oblige a same standard, *secondly* the automated IE are usually expensive algorithms.

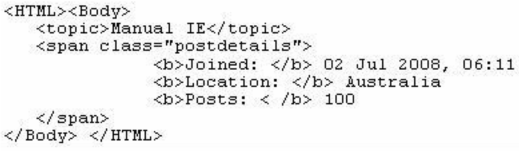
In this paper we described the semi-automatic rule-based Information Extraction algorithm for extracting information from forums for analysis. The rest of the paper is organized as follows; first related algorithms will be discussed in section 2. Section 3 introduces a conceptual framework of the proposed IE engine. Section 4 introduces the theoretical framework of the proposed IE engine. We tested algorithm in section 5 and the results of the tests are outlined in section 6. Section 7 discussed these results and finally we conclude the paper in section 8.

2 Literature Review

Information Extraction in Internet environment is known as a *wrapper*. There are different algorithms for extracting information some of them using HMM (Hidden Markov Model), Naïve Bayesian model or other machine learning model. But generally these algorithms are categorized in three groups as discussed below:

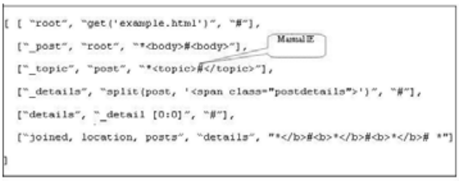
2.1 Manual Approach to IE

Early wrapper generation approaches were done manually [9]. In manual construction of wrapper, programmers analyzed the structure and studied each target website's style. Then, they identified a pattern and programmed a wrapper manually to extract information from these websites. The problem with this approach was that by doing a small change in the style, the pattern required updating manually. Another problem is that it is time consuming and prone to errors, since its manual in nature. With the immense growth of the web and its dynamic nature, it is becoming difficult to maintain the functionality of such wrapper; as a result, there was a shift to look for new technology that can keep up to the pace with which Web was growing. For example, in TSIMMIS project [10] construed manual wrapper has been used to extract semi-structured web pages. This project makes use of a file that specifies where the desired data is located and how the extracted data is packed into an object [10]. The data is extracted based on commands which exist in the specification file. As you see on figure 2, in specification file for extracting desired data from 'example.html' (figure 1), each command is placed in '[]' and include three elements: variables, source, and pattern. As their names suggest, *variables* (*"Joined, Location, Posts"*) hold the extracted results. *Source ("details")* specify the input text to be considered. *Pattern* show how to find the desired data from the source, for example the pattern **<topic>#</topic>* tells that discard everything before the first *<topic>* (* means discard) and then, save in the variables everything after the *<topic>* until the first *</topic>* (# means save). The important operators *split* and *case are used in TSIMMIS.* The *split* divides the input list element into individual elements. User can handle the irregularities in the structure of the input pages by the *case*.



```
<HTML><Body>
   <topic>Manual IE</topic>
   <span class="postdetails">
              <b>Joined: </b> 02 Jul 2008, 06:11
              <b>Location: </b> Australia
              <b>Posts: < /b> 100
   </span>
</Body> </HTML>
```

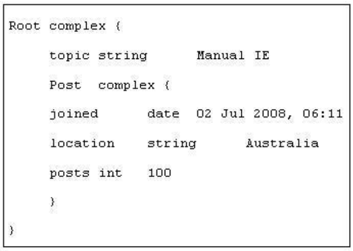
Fig. 1. Example.html



```
[ [ "root", "get('example.html')", "#"],
  ["_post", "root", "*<body>#<body>"],
  ["_topic", "post", "*<topic>#</topic>"],
  ["_details", "split(post, '<span class="postdetails">')", "#"],
  ["details", "_detail [0:0]", "#"],
  ["joined, location, posts", "details", "*</b>#<b>*</b>#<b>*</b># *"]
]
```

Fig. 2. A specification file

Figure 3 shows the TSIMMIS's output.



```
Root complex {
     topic string       Manual IE
     Post  complex {
     joined       date  02 Jul 2008, 06:11
     location     string       Australia
     posts int    100
     }
}
```

Fig. 3. OEM output

We use dynamic algorithm which does not require programming new file for every new style. Our algorithms solves this problem in training phase, more details are outlines in the later sections.

2.2 Semi-automatic Approach to IE

In order to minimize cost and time consumption and maximize accuracy rate of programming, we need a method which generates wrappers in an automatic manner. Machine learning techniques are used to construct wrappers automatically. Wrapper induction [12] is one of these techniques. Induction is the process of reasoning from observed examples to general principles. Figure 4 illustrates a general process of constructing semi-automatic wrapper. In this approach, programmer has to label data items (e.g., a forum's data items: author, posted date, status, message body, and etc. (figure 5)) in a set of pages. In order to reduce intervention of programming skill, a graphical interface is designed for labeling task. A set of rules are extracted from the labeled instances and then the rules are used to extract considered data items from future similar style pages. Human's intervention in semi-automatic method is less than hand-coding method. Humans put their time in labeling instead of coding and debugging. Hence this method is more scalable and can work effectively on a large number of websites.

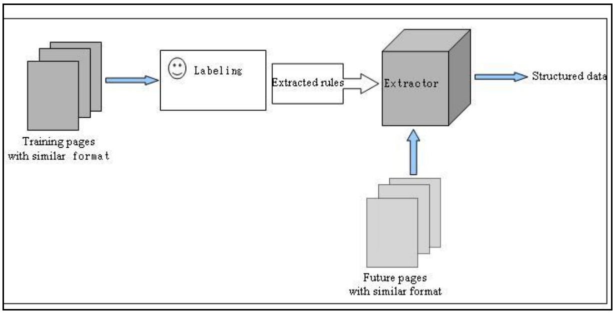


Fig. 4. Proccess of semi-automatic wrapper construction

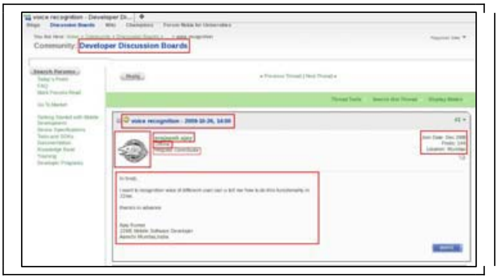


Fig. 5. Forum's data items of www.nokia.com *including c*ommunity's title, discussion's title, author's image, author's name, detail of post, and message body

We now describe two semi-automated systems. The first system is IEPAD an IE system that extracts information from unlabeled Web pages [13]. This algorithm uses rule concept to extract information. It consists of three components: The first component is extraction rule generator which receives an HTML page as an input and transmutes it into a string of abstract representations shown by a binary code with a fixed length. The binary code is passed to the PAT-tree constructor that has the responsibility of building a PAT-tree which is a binary suffix tree. Then the pattern discovers extracts repetitive patterns based on the given PAT-tree and the validator checks the patterns and ignores the undesired ones. Eventually the rule composer revises the patterns and makes regular expressions that are the extraction rules.

The second component is pattern viewer that represents discovered repetitive patterns to the user by GUI and the last component is extractor module that extracts information from similar Web pages based on the extraction rule that is determined by the user. The extractor can use the discovered extraction rule to extract information from other Web-Pages that have the same structure compare to the input webpage. When the extraction rules are discovered the users can select their target pattern and HTML page and send them to the IEPAD, to extract information of the webpage.

The second system is OLERA a semi-automated system that generates extraction rules for unlabeled semi-structured documents [14]. In this system user encloses a block of data that indicates a record's boundary and extraction rule analyze is done based on this data block. This process is done for both singular and multi-record pages in order to find records matched data block. User can have a detailed view of a text fragment or a summarized view of multiple information slots respectively by Drill-down and Roll-up operations. Finally User discards useless information slots and denotes desired ones to specify the scheme of the extraction target. After these operations, extraction patterns are extracted.

2.3 Automatic Approach to IE

However semi-automated reduced maintenance cost, but in large scale with different format, it is still expensive in practice [16]. This approach aims to eliminate manual tasks completely. Analyzing each given format of page and pattern creation is done automatically. In order to generate a wrapper, there is no need to label training examples which teach a wrapper how to extract desired information from future pages. These patterns can provide future pages with similar format. The task that is done by humans is selecting proper pattern, which can extract information of interest from target pages [15]. Automatic wrappers increase the robustness when format of data is changed.

TWWF (Template-independent Wrapper for Web Forums) extract web forums based on visual features such as height, width, font size, content similarity and etc. TWWF has two phases: 1) rendering web forum's information into the DOM tree and then extracting information area from the DOM tree according to the features; 2) identifying posts boundary by dividing the information area into different posts. This approach can just extract posts and replies content but cannot separate other information such as, author, posted date, number of posts and so on [17].

There are a few articles about IE for forums. These are usually about web page ranking for search engine. So, in this section we describe some of these algorithms like Dela and EXALG. This algorithm compares the DOM trees of web pages correspond to the same website and ignores the nodes which have same sub trees. This process is done for extracting data rich sections of a webpage. Second algorithm is EXALG. This method of extraction first receives an unknown template as an input and then finds this unknown template to extract values of encoded pages. This kind of algorithms usually is difficult and expensive in the implementation phase. When we are working in special environment like forums, numbers of template and general view of them is same, hence a semi-automated algorithm is better to use in this algorithm.

3 Conceptual Framework for Automatic Information Extraction

The framework of our IE algorithm contains two steps. We want to extract important information such as author, post content and author number of posts from forums and insert them to database. So this framework can use for forums Information extraction. In this framework base on forum's version different rules are loaded. The detailed framework is described in the next section.

3.1 Solution Overview

As mentioned in above sections, forums are a big source of information which can be used for analyzing the social behavior, search engine, marketing and so many other aims. We need Information extraction methods for extracting process. There is some forums software that people use them like SMF or phpBB or other ones. The developer, user and programmer have to follow a standard. So, we can use semi-automatic rule based extraction to extract forums base on standard that they used. In other words, we can extract a variety of forums just by one rule, because they used one standard. For example, for extracting most of phpBB version 3.0 we need just one rule. But also there are some templates that have different styles which that we should detect them to extract by different rules. A big picture of our system is shown below in Fig. 6:

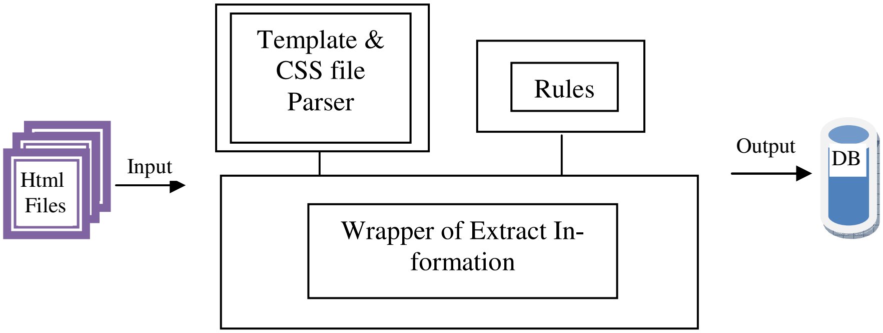


Fig. 6. A big picture of algorithms

This framework has three components. In the Template & CSS file parsercomponent, we want to parse the version of forum and CSS style then send this information to wrapper for extracting, because wrapper needs different rules for extracting information from different forums. In the wrapper component, based on input file and rules that are saved in the database, we configure the wrapper for extracting file base on its information pattern by rules. In the rule component, the rules for different forums are saved and used for retrieval in the extracting process.

3.2 Algorithm Description

In this algorithm, forum page addresses or files are passed as input. This file can be downloaded by web downloader like Teleport. This algorithm needs the file address as input and return the post content, author's username and the author's post number as output. Extraction will be done in 3 steps by our algorithm as described below:

Step 1: Extracting Forum Name & Version

In this step, the forum name and version are extracted. Most of forums use signature that describe the style name and also forum's version. This information is extracted by our intelligent parser. Set of rules are applicable in this scenario that help us to make the right decision.

Step 2: Wrapping Html file

In the second step, specific rule based on step 1 is loaded from the database and generates a parser based on this rule and finally runs the parser to extract information from these files. This information is author's username and message.

Step 3: Insert to Database

In the final step, all extracted information is inserted to the database.

3.3 Algorithm Flow Chart

In this flow chart you can find the algorithm that we discussed in above part, forum's file pass as input to parsers then parsers parse the file to find the forum name and CSS version and call the wrapper by those input (forum name and CSS version). Some forums don't have any signature in html file that introduce forum style and version, so we parse the copy right of website and understand the rule that we should retrieve from DB. Wrapper also loads the related rule from database and start to extract Information and insert them in database. In step 2, rules are loaded from database; each kind of forum may have more than one pattern for IE. You can find the flow chart in below figure 7.

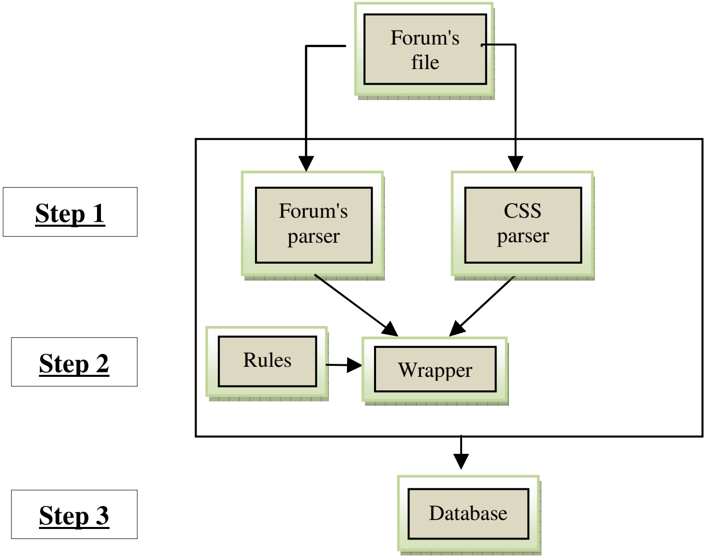


Fig. 7. Flow Chart of the Proposed Wrapper

4 Experimental Setup

We downloaded 72 html files from different forums which used phpBB or SMF forums software, since these are most widely used on the Internet. They had different templates and versions. The list of forums' versions and templates are listed in Table 1. Also number of files with same template and version are listed in Table 1.

Table 1. List of forums, version and template name, total number of templates used for testing

Forum Name	Forum Software	Version	Template	Number of files
The Astro Post	SMF	1.1.10	Standard	10
SoloIngegneria FO-RUM	SMF	1.1.2	Standard	8
www.pietraligure.net	SMF	1.1.4	Standard	2
Parentguideclub	SMF	1.1.6	Standard	1
MERZ ALUMNI EINGETRAGENER VEREIN	SMF	1.1.7	Standard	2
STRABILIARDO	SMF	1.1.8	Standard	3
babonga	SMF	2.0	Standard	4
other	SMF	?	?	3
IREM Houston Member Forum	phpBB	2	subSilver	7
Entertainment comes in blueberry	phpBB	3	Prosilver	17
nukeCops Forums	phpBB	2	Fisubsilver	2
Family / Supporter Support and Resources Forum	phpBB	3	subsilver2	4
other	phpBB	?	?	2

In this experiment we downloaded one page for a specific topic (i.e. from "The Astro Post SoloIngegneria forum" we just downloaded a board by topic "I went to the moon" where one reply was posted) from each forum, where users discussed about a particular topic of interest. Then, we randomly selected 25 topic pages from them for training. In the training stage, rules are identified and extracted and saved in MySQL database. These rules where then used (and further improved), to extract the feature that is listed in Table 2.

Rules are defined to extract 4 main categories as above table. The results of this experiment are shown in next section.

Table 2. List of feature that extract from forums

List of Information Extracted	SMF	phpBB
Author	✓	✓
Number of author post	✓	✓
Author category	✓	✓
Post Content	✓	✓

5 Experimental Results

We have tested our algorithm on72 different forums websites. In other words, we tested our algorithm to extract information from 72 different forums html files. The aim of this experiment was to evaluate the accuracy of the proposed algorithm between different styles and version of forums. It is important to know that each forum has thousands of pages with the same style and template. When we can extract one page of forum successfully we assumed that the remaining pages would be extracted in a similar fashion. Hence we limited our testing to just one page to test the power of rule based algorithms. Our algorithms output is shown in below table (Table 3). Our program can extract most of file successfully nearly achieving 80% success rate. It only couldn't extract 15 files. We are investigating the reasons by improve the results and make is 100% accurate, and this will be our future work.

Table 3. Number of extracted forums, number of forums that algorithms can't extract them

Number of file	Correct extraction	Errors
72	57	15

Rule based algorithms are not flexible by small changes in html file for example in SMF forums if a "<br>" is added after author name our algorithm cannot detect the author's name, but this problem occurs rarely, but it one of the limitations, which we aim to fix in the future. Also in some web sites, administrators remove the forums signature form html file, so in this situation we need to search to find the right style first, for example for phpBB we should search to find "<div class="name">" if we find this pattern we can conclude that this is phpBB.

In this experiment, we use DOM concept for finding rule in html file. But we can develop our work to use DOM concept to draw object tree of forums web page and find the rule from this tree, so for future work we can develop our semi-automatic rule based to automatic extraction algorithm, also the above problems will be solved.

5.1 Experiment One

In the training section, we randomly selected 25 files and labeled information that we wanted to extract from these files (Table 2) then based on these rules are extracted and saved the information in database rule table. We can explain a rule by this example: in phpBB version 2 of subsilver template the post content is between “div” tags, so the rule for extracting post content is “message_tag=”div” message_class=”post””. These rules are different for different forums and style. In the end of training phase, some rules were defined and saved in the rule table. Concurrently, the version of forums and styles were also saved in the forum table. Each rule was related for each forum record in other table. Generally, in the training phase rules, forums version and style name is saved in the relational database.

5.2 Experiment Two

In the test phase, the forum’s file content was processed to find pattern like the rules and extract information from that pattern. For each feature extraction a weight was assigned after the wrapper finished the file processing. We tested our semi-automatic IE for 72 files to evaluate the accuracy. The result showed in Table 3. Most of the errors occurred for files which did not have a rule defined for that template or the template that did not use a standard template. In the figure below the training and test phase and their relationship are described.

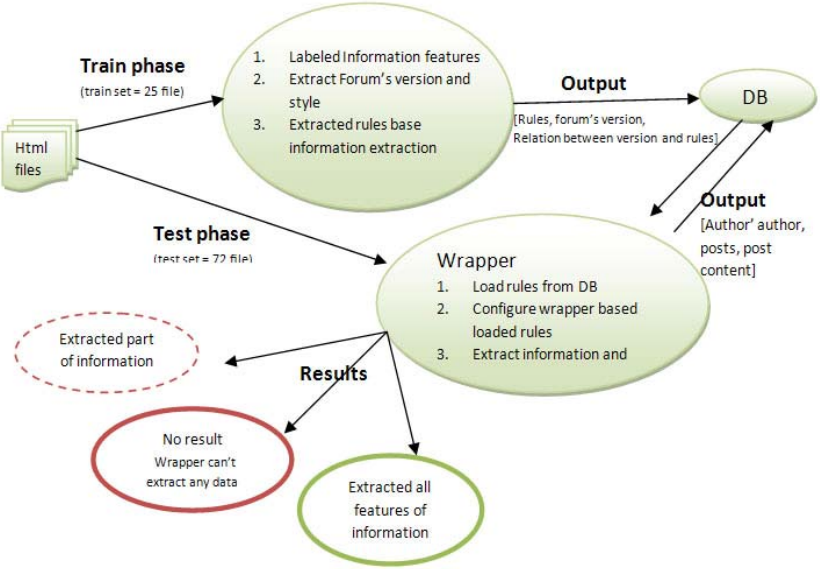

Train phase
(train set = 25 file)
Html files
1. Labeled Information features
2. Extract Forum’s version and style
3. Extracted rules base information extraction
Output
[Rules, forum’s version, Relation between version and rules]
DB
Output
[Author’ author, posts, post content]
Test phase
(test set = 72 file)
Wrapper
1. Load rules from DB
2. Configure wrapper based loaded rules
3. Extract information and
Results
Extracted part of information
No result Wrapper can’t extract any data
Extracted all features of information

Fig. 8. Training and test phases and their relationship

6 Discussion and Conclusion

In this paper we proposed a semi-automatic information extraction algorithm for extracting information from popular discussion boards or forums like SMF, phpBB, vBulletin. We tested the algorithm on SMF and phpBB platforms. A total of 72 different forums were evaluated and the algorithm was tested against this set. Preliminary results indicate promising application of this wrapper. We aim to use this wrapper in the anti-spam technology that we are developing at our Anti Spam Research Lab. In the future we will investigate the possibility of developing a fully automated information extraction algorithm for forums.

References

1. Scardamalia, M.: In: Education and technology: An encyclopedia, pp. 183–192 (2004)
2. PHPBB3, http://www.tesl-ej.org/wordpress/past-issues/volume12/ej47/ej47m2/
3. Welcome to the new Audiworld!, http://forums.audiworld.com/vbguide.pdf
4. Potdar, V., Hayati, P.: Spammer and Hacker, Two Old Friends. In: 3rd IEEE International Conference on Digital Ecosystems and Technologies (DEST 2009), Istanbul, Turkey (2009)
5. Weld, D.S., Wu, F., Adar, E., Amershi, S., Fogarty, J., Hoffmann, R., Patel, K., Skinner, M.: Intelligence in Wikipedia. In: 23rd AAAI Conference on Artificial Intelligence (2008)
6. McCallum, A.: Information Extraction (2005)
7. Iria, J., Ciravegna, F.: Relation Extraction for Mining the Semantic Web. In: Dagstuhl Seminar on Machine Learning for the Semantic Web, Dagstuhl (2005)
8. Kristjansson, T., Culotta, A., Viola, P., McCallum, A.: Interactive Information Extraction with Constrained Conditional Random Fields. In: 19th National Conference on Artifical Intelligence, California, pp. 412–418 (2004)
9. Potdar, V., Hayati, P.: Toward Spam 2.0: An Evaluation of Web 2.0 Anti-Spam Methods. In: 7th IEEE International Conference on Industrial Informatics (INDIN 2009), Cardiff, Wales (2009)
10. Hammer, J., McHugh, J., Garcia-Molina: Semi structured data: the TSIMMIS experience. In: Proceedings of the 1st East-European Symposium on Advances in Databases and Information Systems,
11. Sahuguet, A., Azavant, F.: Building intelligent Web applications using lightweight wrappers. Data & Knowledge Engineering 36(3), 283–316 (2001)
12. Kushmerick, N., Weld, D.S., Doorenbos, R.: Wrapper induction for information extraction. In: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI 1997 (1997)
13. Chang, C.-H., Lui, S.-C.: IEPAD: Information Extraction Based on Pattern Discovery. In: 10th international conference on World Wide Web, Hong Kong, pp. 681–688 (2001)
14. Chang, C.-H., Kuo, S.-C.: OLERA: On-Line Extraction Rule Analysis for Semi-structured Documents. In: 22nd IASTED International Multi-Conference on Applied Informatics (2004)
15. Chang, C.-H., Hsu, C.-N., Lui, S.C.: Automatic information extraction from semi-structured Web pages by pattern discovery, vol. 35, pp. 129–147. Elsevier Science Publishers B. V, The Netherlands (2003)
16. Liu, B.: Web data mining: exploring hyperlinks, contents, and usage data, pp. 323–344. Springer, Heidelberg (2006)
17. Zhang, Q., Yang, S., Huang, X., Wu, L.: Template-independent Wrapper for Web Forums. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pp. 794–795. ACM, New York (2009)

18. Cai, R., Yang, J., Lai, W., Wang, Y., Zhang, L.: iRobot: An Intelligent Crawler for Web Forums. In: 17th international conference on World Wide Web, China, pp. 447–456 (2008)
19. Crescenzi, M., Mecca, V.: Grammars have exceptions. Information Systems 23(8), 539–565 (1998)
20. Hayati, P., Potdar, V.: Evaluation of Spam Detection and Prevention Frameworks for Email and Image Spam - A State of Art. In: Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services, Linz, Austria (2008)
21. Hayati, P., Potdar, V., Talevski, A., Chai, K.: Web Spambot Detection Based on Navigation Behavior. In: 24th IEEE AINA Conference, Perth, Australia (April 2010) (accepted)

Factors Involved in Estimating Cost of Email Spam

Farida Ridzuan, Vidyasagar Potdar, and Alex Talevski

Anti Spam Research Lab, Digital Ecosystems and Business Intelligence Institute,
Curtin University of Technology
farida.mohdridzuan@postgrad.curtin.edu.au,
{v.potdar,a.talevski}@curtin.edu.au

Abstract. This paper analyses existing research work to identify all possible factors involved in estimating cost of spam. Main motivation of this paper is to provide unbiased spam costs estimation. For that, we first study the email spam lifecycle and identify all possible stakeholders. We then categorise cost and study the impact on each stakeholder. This initial study will form the backbone of the real time spam cost calculating engine that we are developing for Australia.

Keywords: spam cost, email spam, spam lifecycle.

1 Introduction

Spamming in email refers to sending unwanted, irrelevant, inappropriate and unsolicited email messages to a large number of recipients. Sending email is fast, convenient and cheap; making it as an important means of communication in business and personal. This is supported by the report from Radicati Group saying that there is a growth of email users from time to time [1]. Dependencies on email usage throughout the whole world provide a huge opportunity to the spammers for spamming.

Spamming activities starts from spammers (who create and send spam), but its impacts goes far beyond them, involving Internet Service Provider (ISP), company, and users (spam email recipients) since they represent the key stakeholders. It is undeniable that each stakeholders involved in this activity has to bear some costs associated with spam.

Throughout our study, there are a few papers discussing on the costs of email spam, but most of them focuses only on one stakeholder, which is the user. Not only that, most of the results are from commercial anti spam vendor. So, it is unclear on how unbiased these reports are. For instance, [2] estimated that company with an average number of employees of 12,000 has to bear the cost of $2.4 million but by deploying the anti spam solution, they would be able to save $1.2 million. Nucleus Research estimated that the loss of productivity for spam management in US is more than $71 billion annually [3]. It is also estimated in [4] that deploying Spamhaus in large corporation and mid-sized corporation could save $400,000 and $27,000 respectively.

Therefore, the main aims of this paper are to 1) identify spam stakeholders, 2) understand email spam lifecycle, 3) identify cost categories and parameters for each cost

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 383–399, 2010.

categories, and 4) derive the cost impacts based on the identified cost categories and related parameters towards each stakeholder.

This paper has been organized in the following way. Section 2.0 will enlist three main stakeholders in email spam lifecycle i.e. spammers, ISP and users. Section 3.0 will give a brief overview of the lifecycle of email spam. Details on which party involve in every stage, tools used by the spammers are also included in this section. Section 3.0 continues by introducing 5 cost categories and its related 17 parameters that can be used to estimate the cost of email spam. Section 4.0 begins by laying out the email spam costs in detail towards three different stakeholders: spammers, users and ISPs. The last section provides the discussion and conclusion.

2 Spam Stakeholders

In this section, we list all the key stakeholders in the email spam lifecycle. These include

=> Spammer
=> Internet Service Provider (ISP), and
=> User

We now explain the details of each stakeholder in the following sections. We specifically outline where each stakeholder plays a key role.

2.1 Spammer

Spammer starts the lifecycle of email spam by creating spam messages and sends them to the users. Spammer uses various techniques and tries to bypass filters deployed by other stakeholders. Spamming gives a few benefits to the spammers such as to generate revenue, get higher search rank, promote products and services and others [5], which motivates them to continue spamming even with the existence of spam laws such as the CAN-SPAM Act [6].

2.2 Internet Service Provider (ISP)

Internet Service Provider (ISP) provides internet access both to users and spammers. They are involved in the lifecycle of email spam because without their service, spammers would not be able to send emails in bulk. On the other hand, emails users would not be able to read their legitimate email without ISP. Due to the spamming activities by spammers, ISP would have to prepare large bandwidth for their users, which indirectly increases ISP's operational costs.

2.3 User

The third stakeholder in the lifecycle of email spam is user. User is the actual recipient of an email spam. Spammer's goal is that the email should reach the end user and ensure that user is interested in opening, reading and responding to the email. Apart from the filtering system, the end user is the key to decide whether the spammer's campaign would be successful or not. This is because, in the end, if there is any spam email that gets to the inbox; the user can choose either to respond or ignore that email. The lifecycle ends when the user ignores the spam email but it continues if the user replies or takes action based on that email. Users depend on efficiency of the anti spam software to avoid getting spam emails. Most of the problems faced by the users are quite similar. They are afraid of losing legitimate emails filtered by anti spam filter but at the same time, they do not like to spend too much time checking spam emails. The user may even waste more time if s/he gets interested in a spam email spam because they will spend more time browsing unnecessary websites.

3 Lifecycle of Email Spam

The three main stakeholders in email spam lifecycle are users, spammers and ISPs. Based on these stakeholders we have classified email lifecycle into seven main stages. Email spam lifecycle starts from spammer's end and then continue to traverse to the recipient's end. We categorise email lifecycle into the following seven stages as follow:

=> Get email addresses
=> Create spam messages
=> Send spam
=> Filter spam by the ISP
=> Filter spam on server side
=> Filter spam on client side
=> Spam that bypasses all filters [7, 8].

As mentioned earlier, this section will focus more on the lifecycle of email spam, tools used by spammers, problems encountered by spammers in sending the spam and what has been done by anti spammers as countermeasure in each stage. The first three stages involve spammers while the next four stages involves with the recipients of spam email. These seven stages are described in the figure below with three stakeholders involved: users, spammers and ISPs.

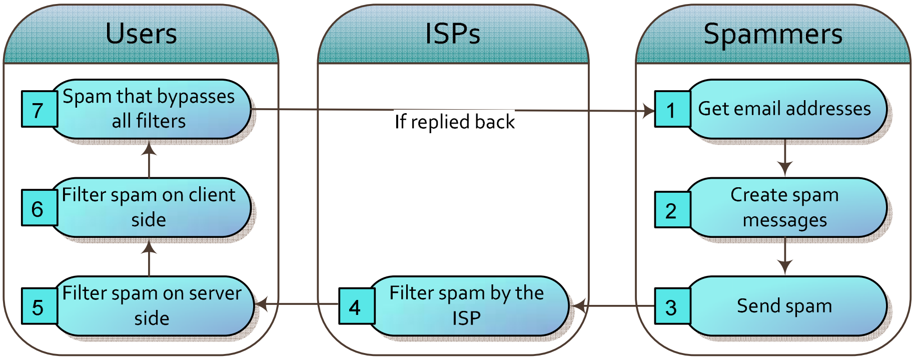


Fig. 1. Lifecycle of Email Spam

3.1 Get Email Addresses

The lifecycle of an email spam starts with the spammers gathering a list of email address. These email addresses can be gathered through several well known ways such as obtaining email list & newsletter subscriber list through hacking or buying it, using spam bots to crawl and collect email addresses from websites, randomly generating name combinations for certain domain and others. Several other ways of spammers getting email addresses are explained in [9]. In fact, in a recent case, spammers were successfully in getting a huge list of email addresses with its passwords, that significantly increased the rate of spam [10] since they are able to manipulate the account itself and use it in order to get other active email addresses.

One of the easiest ways for spammers to gather email addresses is to use automated tools e.g. Speed Email Extractor, Power Email Harvester, Email Grabber, Teleport Pro, Email Spider or Email Extractor from EmailSmartz and others [11-15]. Trial and limited version are downloadable for free. Some of this software such as Power Email Harvester [12] and Email Extractor [15] could even be used to send bulk mail.

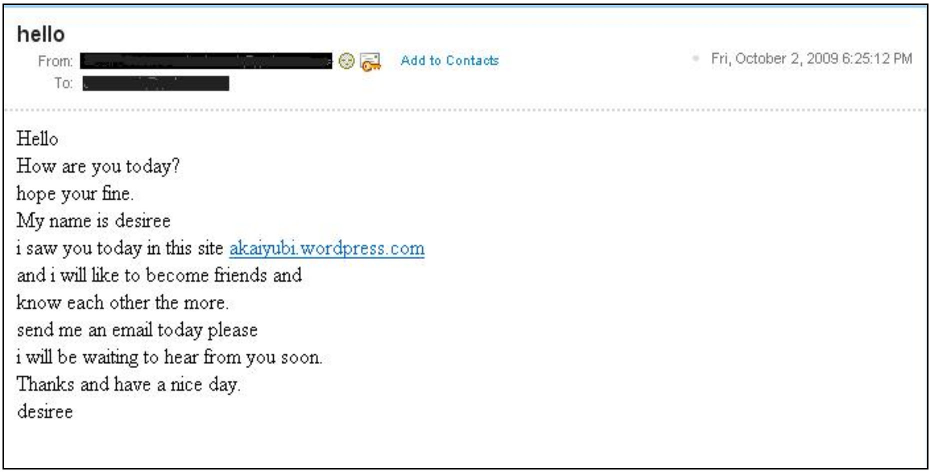
There are plenty of ways for spammers to gather a list of email addresses. Nevertheless, it would be wasteful if the list that they have gathered contains fake email addresses. Spammers need to test the list of email addresses by sending an email to detect whether it's fake or not. If it is, spammer usually gets an auto reply message saying that the address is not valid. Still, this does not cost much for spammers to stop spamming. In order to maintain a genuine email database, there is no other way than to keep updating the database and getting a new list.

Preventing spammers to successfully gather email addresses, companies need to deploy security measures on network server to prevent spammers from hacking the server. In case where spammers use crawlers to obtain email addresses from websites, web administrator could use tools like Spam Preventer 1.0 [16] to avoid websites from being harvested by spam bots. Nowadays, users also have been educated not to simply put their email address on the websites and use address munging technique such as by replacing "[dot]" instead of "[.]".

3.2 Create Spam Messages

Stage 2 shows that spammers need to create messages before sending the email. It is possible to just send a simple message to thousands of users but the rate of success for such emails being read is low. Research is needed to provide users to read what they need [17] so that spammer could sends a specific email to their target group. We believe that spammers have their own database of words and phrases to create messages and this system would allow them to create messages and send huge amount of spam messages faster. This system is supposed to be robust against simple keyword filter.

Common users would have to follow certain tips to avoid from being mistakenly seen as spammers, such as given in [18-20]. Similarly, spammers could also follow these tips in order from being detected. Hence, spammers will try to imitate real user's behavior. Their target is to get users interested in reading the messages and make an active action either by ensuring that the user clicks on the link provided or reply to the email [17]. Nowadays, spammers are smart enough to avoid using common keywords since they are easily detectable. Example below shows that spammers are now trying to act as if the spammer knows the user personally.



hello

From: [redacted]

To: [redacted]

Fri, October 2, 2009 6:25:12 PM

Hello
How are you today?
hope your fine.
My name is desiree
i saw you today in this site akaiyubi.wordpress.com
and i will like to become friends and
know each other the more.
send me an email today please
i will be waiting to hear from you soon.
Thanks and have a nice day.
desiree

Fig. 2. An example of smart spam message

Spammers would try to avoid using sensitive blacklisted keyword and exploit the spelling of those words to bypass filters. Spammer also uses personal words such as “Hey” or “Hello” such as in Figure 2 as their message title to encourage users to open the email. This makes it difficult for anti spam filters to differentiate between spam and ham (i.e. legitimate) messages. Nevertheless, there is no other way for anti spam filters that uses content based method to keep updating their keyword databases. Hence, using behavior analysis along with content based analysis might provide a better solution [21].

3.3 Send Spam

Spammers then will send the created messages to thousands of recipients. There are several ways for spammers to do this. The most important thing for spammers is to ensure that their identity remain unknown when sending spam. Spammers might use various different free email services provided by Hotmail, Yahoo! or Google so that they cannot be detected by the recipient. The spammers might also use unsuspected third party mail servers and spam can be relayed through them indirectly hiding the spammer’s identity. Spammer could also scan for open insecure proxy servers or botnets, which can be used to collect email addresses and then send an enormous amount of email simultaneously.

Spammers then create or use program or software that could be used to send spam to the list gathered earlier in a faster way. Some of the tools that can be used to send email in a huge volume or mail bombers are Bulk Mailer, Avalanche, Dark Mailer, PostCast Server [22-25]. Meanwhile, in order to detect insecure proxy, spammers could use YAPH (Yet Another Proxy Hunter) [26] or Proxy Hunter [27]. An easier way for spammers to send email is to use email sending company such as Aweber Communications, Constant Contact and JangoMail [28-30].

On the other side, if the spammers are using open proxy server, there might be other spammers that are spamming from similar proxy as well. This could make it obvious for others that this particular server is used to send spam hence shortening the life span of using that proxy from being banned. In this way, spammers then need to find a new open proxy. Spammers could also be sending spam email through email advertising companies, which are sending a numerous amount of email legally. The price that spammers have to pay for an email is also comparatively very low [31]. Nowadays, using botnets is one of the easiest and effective ways to send spam. Nevertheless, there are some botnets that have been detected such as Srizbi, Bobax and Rustock [32]. Still, spammers could just find other botnets that have not been detected and continue spamming.

In this case, the anti spam group just have to keep detecting active botnets that are sending huge amount of spam email and blacklist them. Companies should also keep updated list of banned hosts and flag suspicious IP address. To avoid company's server from being used by spammers, company needs to deploy anti relay functionalities on their servers. It is without a doubt that more active countermeasure needs to be taken in order to avoid spammers from keep spamming.

3.4 Filter Spam by the ISP

Once spammers send spam email, those spam messages will first be filtered by the ISPs. These anti spam filter tools usually implement Sender Reputation, Sender Authenticity, Content Analysis (though some of the service provide claimed that they are implementing a better solution than this method, but it is unclear of what method they are using), Network Analysis and others. The pricing of these tools' depends on the number of mailboxes that the ISP wishes to protect, numbers of messages to be handled daily and type of institution. Some of the tools cost depends on the number of domains, or number of servers.

Tools that can be deployed by the ISP are MailCleaner, SpamFilter ISP from Logsat Software and SpamTitan [33-35]. Several other choices for the ISP are the Anti Spam for ISP by Kaspersky and MXForce [36, 37]. As of November 2009, ovh.net, telefonica.es and tiscali.it are some of the top three spam service ISP reported by Spamhaus [38]. Some other services to detect and blacklist ISP used for spamming are RBL(Real Time Blacklist) such as provided by [39] and [40].

ISP on the other hand is spending a huge amount of money for the anti spam filters [41, 42]. ISP also needs to control and manage this tool. At the same time, they do not want to lose their customers, which may also include spammers. It is also believed that there are ISPs that sell services to spammers in order to gain profit [38]. This is one of the reasons why spammers are unstoppable.

3.5 Filter Spam on Server Side

Company usually deploy anti spam filter on their server. Therefore, users would not see all the emails that were actually sent to them because most of these email messages have been filtered by the server. The cost of implementing anti spam filter on the server depends on several factors such as how many email can the tools cater at a time, methods and additional services provided by the anti spam filter, number of users, duration of license and others. These filters usually provide solution by using Bayesian filtering, keyword checking, email header analysis and DNS blacklist [17].

Some of the commercial tools that could be implemented on the server side include EMP 7 Enterprise Anti Spam, SpamFighter Exchange Module, Spamfighter Mail Gateway and Web & Mail Security GFIMailEssentials Anti-spam Solution for Exchange/SMTP/Lotus [43-46]. In addition, companies could also opt for open source tools such as SpamAssassin and Anti-Spam SMTP Proxy Server [47, 48].

Main problem for applying filter on server side are the costs of renewing the license every year and managing the tools itself especially for a small company. Company usually does not favor to commit for a longer duration license because the service provided might not satisfy the company and there might be better services provided by other anti spam companies in the future.

3.6 Filter Spam on Client Side

As an additional precaution, users themselves can install desktop application to filter spam. Tools that can be used by the users are BullGuard Spamfilter, Cloudmark Desktop, ClearMyMail, MailFrontier Desktop, Spam Filter Express and others [49-53]. These tools come with various features and different advantages and disadvantages. Some tools need to be updated frequently; some are licensed, which has to be paid every year and some need more time to train.

With this filter, it is much harder for the spam messages to get through to the users. But still, not everybody would want to spend money on additional filter. Some users are just satisfied with what is provided by the application. The logic comes when users use free email services, therefore, they would not want to spend more on additional filtering and just hope the free services that they chose provide the best anti spam technology. These are the groups that spammers usually choose to spam. Even for the users that implement this filter, spammers could just hope that they will still check the spam folder and response to the spam message.

As for users, the problem comes when receiving smart spam messages. Users without knowledge could easily fall for the trap especially with spammers imitating real friendly behavior. Still, the tool used to filter spam on client side usually comes with a high costs and need to be maintained personally by user.

3.7 Spam That Bypasses All Filters

Spam that is not caught by both filters then can be read by the users. If the users fall for the trap and reply that email, user's email address is confirmed to be active and will then go into spammers' email database. As a result, user then will receives more email spam. The only problem for spammers is that most of the spam messages sent by them are already filtered. Hence, they are targeting specific topic or product to specific person in order to get them to read the messages. As a countermeasure, knowledge of what spam is should be given to educate email users from being easily manipulated.

4 Cost Categories of Email Spam

We have identified 5 cost categories with 17 parameters to estimate the cost of email spam. This section will further provide description of each cost category with its associated parameters. Parameters in each cost category will be defined considering that we are collecting a huge collection reference of web spam and trying to estimate those cost based on the collection.

4.1 Storage Cost

Storage cost refers to the cost spent for server storage used to store any information such as list of email addresses, spam for spammers and blacklisted IP addresses for companies and ISPs. In the effort to avoid losing legitimate messages, it is easier to flag an email or spam content so that it could be checked by the user itself. Once the user checks it, the user either would read it and clear the messages as non-spam or delete it if it is a spam. This process requires additional storage and includes the cost of filtering because the efficiency of this method depends on the filter itself. Suppose the email or content itself contains big attachment files or large sized images, this will increase the storage requirement and its cost. Storage cost for email spam can then be defined as follow:

$$C_s = f(a,b,c,d). \tag{1}$$

where

a = monthly fee/ GB for storage,
b = spam message received/day,
c = message size,
d = duration of storage.

Monthly fee per GB for storage is actual server cost paid for each GB. Measurement of this cost will consider the current general cost of storage. Spam message received per day is defined as spam messages received by the user per day. Measurement of this unit will consider all spam emails received by all the users on a certain period.

Message size is then defined as size of spam email. Measurement of this unit will consider the actual size used to keep the spam email message in storage. Duration of storage is the parameter defined to calculate how long (days) a message is stored before it is checked and deleted. In order to measure this, there is a need of a close observation towards the spam email to measure when a spam email is checked and deleted since it was first received.

4.2 Bandwidth Cost

Bandwidth cost is the cost used for connectivity. In this case, all parties are going to bear the cost of i.e. spammers for spamming and users for checking emails. Bandwidth cost function for email spam can then be defined as follows:

$$C_b = f(e, f, g). \tag{2}$$

where

e = annual fee for connectivity,
f = email percentage representing bandwidth,
g = spam percentage of all email.

Annual fee for connectivity is the actual cost users have to pay for Internet connection. Measurement of this cost will consider the current cost that users have to pay for the connectivity. On the other hand, email percentage representing bandwidth is considered as the proportion of bandwidth used just for email purposes. Measurement of this cost would need to consider previous research done by [3, 8]. Spam percentage of all email is defined as the proportion of spam messages from all received emails. Based on data collection, the proportion of spam email messages can be measured based on the storage that it uses.

4.3 Human Resource Cost

Human resource cost for spam filter is the cost used by the associated party for filtering spam. Considering that not everyone has knowledge of spam, companies usually hire a professional team to handle any issues arising from spam. This could include help-desk support or network team specially hired for fighting spam. Spam sometimes cause serious problem if they are embedding virus or worms with the attachment [7]**.** Ignoring the cost associated with virus and worms, the cost for human resource can be defined as follow:

$$C_{hr} = f(h). \tag{3}$$

where

h = salary for human resource incharge to support spam.

Salary for human resource incharge to support spam is the amount of salary paid for person to manage spam-related problem. Measurement of this cost depends on the current salary usually paid to the network administrator, support team members or help-desk officer, which requires further survey on current situation in order to determine its precise and accurate value.

On the other hand, spammers are also using their time to create smart spam, find a list of genuine email addresses and send spam etc. This cost is associated from stage 1 to stage 3. In this case, we define human resource cost for spammers as follow:

$$C_{hr} = f(i). \tag{4}$$

where

i = time used for spammers to spam.

4.4 Annual Productivity Cost

Three different substance to take into account when calculating productivity cost are the 1) process of inspection and deletion of spam that gets through to the inbox, 2) process of identifying legitimate email from spam folder and 3) helpdesk support [54]. In our case, annual productivity cost is measured for the time that is spent on each spam message and the cost of helpdesk support is already calculated in human resource cost. This cost may vary depending on user's knowledge. Even if a spam message is flagged, a user might actually reply the email. This is because a message might be spam to one, but it might not be for others. Nevertheless, this cost can be defined using several parameters as follow:

$$C_{ap} = f\left(j, k, l, m, n, o, p\right). \tag{5}$$

where
j = time to clear out spam/each check,
k = time to look for false positive in spam folder/each check,
l = time to focus back on work after each check,
ml = employee salary,
n = how many times users check email/day
o = how many working days
p = number of employees in one organization

Time to clear out spam/each check is considered as the amount of time needed to delete any spam email. Measurement for this cost depends on how fast a user can interact with system which also depends on how familiar users are with the application.

Time to look for false positive in spam folder/each check is defined as the amount of time needed to check and determine if there is any a legitimate content that was mistakenly flagged as spam. Measurement for this cost may vary depends on how knowledgeable users are about spam. It is also possible to measure this based on author's experience.

Time used to focus back on work after each check is the parameter used to define the amount of time needed to an employee to focus back to work after each check. This cost is usually measured by taking an average value of user opinion. Measurement for this cost has not been decided yet but it is also possible to measure this based on author's experience.

Employee salary is the parameter that would consider the salary of an employee which varies depends on position hence having a different effect on the total amount of this cost. Measurement for this parameter need further survey on current situation.

How many times users check email/day is considered as the frequency of a user checking the application. It is also possible to use a predetermined default value for this parameter.

For parameter how many working days, it is possible to just use a predetermined value that is 22 days permonth considering that there are 30 days in every month.

Number of employees in one organization is the parameter could be measured through the number of account holders for email application. These parameters would play an important role in calculating the cost of software because some software are licensed and buying them depends on the number of employees in an organization.

4.5 Software Cost

There are a lot of software tools available for both email spammers and users. Considering that each party deploys these tools for their email application, it is important to measure this cost based on current survey of the cheapest and most effective spam tools for spammers and anti spam solutions for users. This cost can be defined as follow:

$$C_{sw} = f(q). \tag{6}$$

where

q= software costs.

Table 1 shows that parameters used in cost calculation for each party i.e. spammer, company and ISP. Further explanation on each cost calculation for all the parties will be provided in the next section.

Table 1. Parameter used for spammer, company and ISP

	Parameter	Abb.	Spammer	Company	ISP
Storage Cost					
1	Monthly fee/GB for storage	*a*	✓	✓	✓
2	Spam message received/day	*b*		✓	✓
3	Message size	*c*	✓	✓	✓
4	Duration of storage	*d*	✓	✓	✓
Bandwidth Cost					
5	Annual fee for connectivity	*e*	✓	✓	✓
6	Email percentage representing bandwidth	*f*		✓	✓
7	Spam percentage of all email	*g*		✓	✓
Human Resource Cost					
8	Salary for human resource in charge to support spam	*h*		✓	✓
9	Time used for spammers to spam	*i*	✓		
Annual Productivity Cost					
10	Time to clear out spam/each check	*j*		✓	
11	Time to look for false positive in spam folder/each check	*k*		✓	
12	Time to focus back on work after each check	*l*		✓	
13	Employee salary	*m*		✓	
14	How many times users check email/day	*n*		✓	
15	How many working days	*o*		✓	
16	Number of employees in one organization	*p*		✓	
Software Cost					
17	Software costs	*q*	✓	✓	✓

5 Spam Cost Impact towards Stakeholders

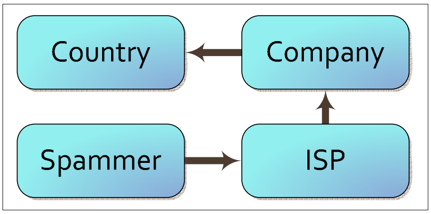


Fig. 3. Email Spam Cost Impacts

Figure above shows the cost impact of email spam towards four parties: spammer, company, ISP and country. Based on our understanding, spammers relatively bear the lowest cost followed by company, ISP and country. However, in this section, we are only going to focus the cost impact of spam towards three stakeholders that we have defined in Section 2. For this section, we are going to address company as a stakeholder since company is considered as a group of users. Based on the generic parameters set in previous section, each cost associated for spammer, company and ISP are going to be identified.

5.1 Cost of Email Spam for Spammer

Based on the lifecycle that we have mentioned earlier, storage cost is associated with spammers in stage 1 and 2 and with users in between stage 4 and 5. Hence, parameter set for storage cost and bandwidth cost used by spammer are as follow:

$$C_s = f(a, c, d). \tag{7}$$

$$C_b = f(e). \tag{8}$$

With free web crawler, spammers can use UbiCrawler, WebSphinx, Wget, Polybot and Teleport to gather a list of email addresses [14, 55, 56]. In this stage, spammer could use email grabber such as Speed Email Extractor, Power Email Harvester, Email Grabber, Teleport Pro, Email Spider or Email Extractor from EmailSmartz and others for free [11-15]. Usually, the latest version of the software which provides more functionality would cost the spammers in the range of AUD19($16.95) to AUD165($149.95). Nevertheless, storage cost in stage 1 for spammers is relatively low as they can just use normal capacity to store a list of billions email addresses. Thus, the only cost that spammers have to spend is their time as the software that they bought is a one-time cost. Time uses for spammers to spam is defined as in Equation 4.

Some of the tools that can be used to send email in a huge volume or mail bombers are Bulk Mailer, Avalanche, Dark Mailer, Direct Mailer which costs spammer in the range of AUD45($40) to AUD550($499). YAPH(Yet Another Proxy Hunter) which is an open source or Proxy Hunter for AUD33 ($29.95) are tools that can be used to detect insecure proxy servers. Spammers that generate high revenue could also opt by sending email using companies such as Aweber Communications, Constant Contact and JangoMail [28-30]. These companies provide services which would cost spammers as low as AUD0.0044 per email recipient.

5.2 Cost of Email Spam for Company

Regardless of how users or companies deal with spam, there are costs that they have to bear and these costs are far beyond than financial costs. For example, according to Nucleus Research, there are three ways used by a company to deal with email spam which are: confirmation process, quarantine strategy and delete strategy [3]. Each strategy possesses different risk thus causes different costs such as loss of productivity, loss of time and storage cost.

Storage cost is associated between stage 5 and 6 for company. Storage cost parameters for company can be defined similarly as in previous section. Using the quarantine strategy, all email marked as spam needs to be kept in storage until the users check and delete it. This process is implemented in order to avoid losing important business emails.

Bandwidth is a valuable resource for company. Now that spammers are getting smarter and more creative by embedding various attachment types, downloading all these unnecessary spam messages consumes large bandwidth as well. Bandwidth cost is associated during the transmission of email spam from spammer's side to user's side which is between stage 3 to 5 and this cost can be defined as in previous section.

Email provides a faster and smoother communication approach, but when used by spammers, it could turn its advantages towards being one of the reason contributing to a big cost for a company. Company need to spend their money on spam filtering tools, hire related personnel to deal with this problem and even provide training for their employees to improve their understanding of spam. Support cost for spam filter is associated with stage 5 and 6 in the case of lifecycle of email spam and can be defined as in Equation 3.

Employees also need to spend time in checking spam folder to avoid losing legitimate messages. They then need to read and delete or mark as non spam for each email messages for future safety. This situation worsen if the employee decide to reply or get interested in the product or services by the spam messages which lead them to spend more time browsing unnecessary websites. They also usually take time before getting back to do their work. This affects the employees' productivity [57, 58]. This cost is associated with stage 5 and it is calculated as in Equation 4.

As far as software costs goes, open source tools that can be deployed in company's server are SpamAssassin and Anti-Spam SMTP Proxy Server. Conversely, there are commercial tools which are EMP 7 Enterprise Anti Spam, SpamFighter Exchange Module which could costs AUD40($36) per user, Spamfighter Mail Gateway with AUD22.50($20.40) per user and Web & Mail Security GFIMail Essentials Anti-spam Solution for Exchange/SMTP/Lotus with 10 to 24 Mailboxes at a price of AUD43.05 per mailbox.

Based on our observation, companies would have to bear the cost of buying one license for every single user and this cost usually gets cheaper if they buy more licenses. In this case, a small company would have to pay a higher price for anti spam tools. For example, the cost of using EMP 7 Enterprise Anti Spam is AUD28.60 ($25.99) for every user. This cost gets cheaper if the company buy the license for more users that is AUD10.60 ($9.60) for each email recipient for 500 to 999 users. For additional filtering, users themselves could equip their desktop with anti spam filter tools which could cost them below AUD55 (US$50).

5.3 Cost of Email Spam for ISP

ISPs need to spend additional costs for storage and bandwidth to provide services to their client who can be spammers or normal users. Spam which is transmitted at the same time with legitimate content causes increase usage of network bandwidth and storage capacity. This cost parameters are similarly defined as in Equation 1 and Equation 2. ISP could also deploy anti spam tools such as MailCleaner, SpamFilter ISP from Logsat Software which cost AUD660 ($600) per server. Another tool called SpamTitan with single appliance license for 50 users covered for 1 year could be bought at a price of AUD435 ($395).

6 Discussion and Conclusion

Symantec reported that the average of spam volume is 87% of all email messages[59]. Another firm, Ferris Research in their recent report estimated the cost of email spam for the whole worldwide is $130 billion[8]. It shows that spam impacts are not limited to an individual but to a certain degree it could affects a country. Several figures have been produced by Japanese researchers showing that a huge amount of money was spent or wasted due to spam mail. It is reported that 960 billion yen was calculated for GDP loss from several big industries in Japan [57].It is a great loss suffered just to pay for time spent in handling email spam and the labour needed in order to process spam mails. This labour loss is associated with time spent for email spam such as what we have defined in Equation 4. It is proved by [57, 58, 60] that email spam harm the economy of a country which without proper effort could further reduces the economic growth globally.

This paper first identifies the lifecycle of email spam and the cost of spam associated with each lifecycle stages. Considering that we are going to measure the amount of spam accurately based on a huge reference of spam collection, there is a need to formulate all associated costs accordingly which was done in Section 5 where cost categories have been defined with 16 related parameters.

Regardless of facing all these costs, it is important to take note that there are several key issues in calculating the cost of spam. A considerable amount of report has been published on the cost of email spam. Nevertheless, there is no guarantee that surveys or report done are unbiased as most of the report will finally try to show that the cost of spam can be reduced by using their product, hence it is also possible that in earlier stage, they would try to maximize the cost of spam.

This paper provide an overview of three different stakeholders bearing spam costs including spammer's cost to spam, ISP's cost and company's cost in combating spam. We are trying to define a much more general way in calculating spam costs in a case where a huge reference real data collection is done. As a conclusion, it is important to continue on researching on real time spam cost calculator in order to ensure that company are spending a considerable amount of cost in combating spam. By studying cost of spam, it is hoped that spammer's in future would have to spend more than the amount that they gain so that they would lose their interest/benefit for spamming.

References

1. Number of e-mail users worldwide to reach 1.6 billion in (2011), says Radicati Group, http://software.tekrati.com/research/9512/ [cited November 15, 2009]
2. Windows & .NET Magazine, The Secret Cost of Spam
3. Nucleus Research, Spam: The Repeat Offender, in Research Note (2007)
4. Osterman Research Inc., How Spamhaus Cost-Effectively Eliminates Spam, in An Osterman Research White Paper (March 2008)
5. Hayati, P., Potdar, V.: Evaluation of spam detection and prevention frameworks for email and image spam: a state of art. In: Proceedings of the 10th International Conference on Information Integration and Web-based Applications \& Services. ACM, Linz (2008)
6. PUBLIC LAW 108–187—DECEMBER 16 (2003), http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=108_cong_public_laws&docid=f:publ187.108.pdf [cited November 15, 2009]
7. Nagamalai, D., Dhinakaran, B.C., Lee, J.-K.: An in-depth analysis of spam and spammers (2009)
8. Ferris Research, Spam, Spammers and Spam Control, Ferris Research: San Francisco, Cali. USA (March 2009)
9. Raz, U.: How do spammers harvest email addresses (2009), Available from: http://www.private.org.il/harvest.html [cited October 28, 2009]
10. Lanxon, N.: More email passwords posted to the Internet: Experts detect spam increase (2009), http://crave.cnet.co.uk/software/0,39029471,49303853,00.htm [cited October 20, 2009]
11. Download3000. Speed Email Extractor 6.0 Free Download, http://www.download3000.com/download-speed-email-extractor-count-reg-44069.html [cited November 13, 2009]
12. Brothersoft. Power Email Harvester 1.45 Download. http://www.brothersoft.com/power-email-harvester-86025.html [cited November 13, 2009]
13. Emailgrabber.net. Email Grabber, http://www.emailgrabber.net/ [cited November 13, 2009]
14. Tenmax.com. Teleport Pro - Offline Browsing Webspider, http://www.tenmax.com/teleport/pro/home.htm [cited November 13, 2009];
15. E-mailSmartz. Email Marketing Software | Email Generator, http://www.emailsmartz.com/ [cited November 13, 2009]
16. Spam Preventer 1.0., http://wareseeker.com/Security-Privacy/spam-preventer-1.0.zip/346607 [cited November 13, 2009]
17. Spammer, X., Posluns, J., Sjouwerman, S.: Inside the SPAM Cartel: By Spammer-X. Syngress 2004

18. San, G.: Creating Email Copy - Spam Words to Avoid, http://ezinearticles.com/?Creating-Email-Copy-Spam-Words-to-Avoid&id=2510485 [cited November 24, 2009]
19. How to stop yourself from being unfairly labelled a spammer, http://www.emailaddresses.com/email_spam_lists.htm [cited November 24, 2009]
20. Connick, E.: How to stop your e-mail from being seen as spam, http://www.helium.com/items/1063324-how-to-stop-your-e-mail-from-being-seen-as-spam [cited November 24, 2009];
21. Hayati, P., Chai, K., Potdar, V., Talevski, A.: HoneySpam 2.0: Profiling Web Spambot Behaviour. In: Yang, J.-J., Yokoo, M., Ito, T., Jin, Z., Scerri, P. (eds.) PRIMA 2009. LNCS, vol. 5925, pp. 335–344. Springer, Heidelberg (2009)
22. Brothersoft. BulkMailer 2.3 Download, http://www.brothersoft.com/bulk-mailer-16936.html [cited November 13, 2009];
23. Wareseeker. Avalanche 98.8.20., http://wareseeker.com/Email-Tools/avalanche-98.8.20.zip/4692 [cited November 13, 2009]
24. Brothersoft. DarkMailer 1.13 Download. http://www.brothersoft.com/dark-mailer-90307.html [cited November 13, 2009]
25. Postcast Server. PostCast Server - Free SMTP Server, http://www.postcastserver.com/download/ [cited (2009), 13 November 13, 2009]
26. Sourceforge. YAPH - Yet Another Proxy Hunter for HTTP Connect, Socks4 and Socks5 Servers, http://yaph.sourceforge.net/ [cited November 13, 2009]
27. Hunter, P.: http://www.proxyblind.org/proxy_hunter.shtml [cited November 13, 2009]
28. AWeber Communications. Email Marketing Software, Email Newletters and Autoresponders by AWeber, http://www.aweber.com/ [cited November 14, 2009];
29. Constant Contact. Email Marketing Solutions from Constant Contact, http://www.constantcontact.com/index.jsp [cited November 14, 2009]
30. JangoMail. Email Marketing & Personalized Business Email Delivery Service, http://www.jangomail.com/ [cited ovember 14, 2009]
31. Judge, P.: The state of the spam problem. Volume (2003)
32. Stewart, J.: Top Spam Botnets Exposed. April 8 (2008), http://www.secureworks.com/research/threats/topbotnets/ [cited October 29, 2009]
33. Mailcleaner Anti spam solution for enterprise or ISP, http://www.mailcleaner.net/ [cited November 14, 2009];
34. LogSat Software. Spam Filter ISP - spam filter server for Windows | Spam Filter ISP, http://www.logsat.com/ [cited November 14, 2009]
35. SpamTitan | Leading way to better e-mail security, http://www.spamtitan.com/ [cited November 14, 2009]
36. Kaspersky Lab. Kaspersky Anti-Spam ISP Edition, http://www.kaspersky.com/corporatesolutions?chapter=4157640 [cited November 14, 2009]
37. MX Force: Managed Email Security Services, http://www.mxforce.com/ [cited November 14, 2009]
38. Spamhaus. Spamhaus Statistics: The Top 10. The 10 Worst Spam Service ISPs (2009), http://www.spamhaus.org/statistics/networks.lasso [cited October 28, 2009]

39. Spam Blacklist Checker, RBL Black listed IP address, blacklist, check blacklists, http://www.spamblacklist.com.au/ [cited November 14, 2009]
40. RBL.JP, http://www.rbl.jp/ [cited November 14, 2009]
41. Sipior, J.C., Ward, B.T., Bonner, P.G.: Should spam be on the menu? Communications of the ACM 47(6), 59–63 (2004)
42. Atkins, S.: Size and Cost of the Problem. In: 56th IETF Meeting, San Francisco, CA (March 2003)
43. Korsmeyer. Enterprise Email Security for Exchange Server, Domino and GroupWise, http://www.jak.com/ [cited November 14, 2009];
44. Wareseeker. SPAMfighter Exchange Module 3.5.0.0. http://wareseeker.com/Email-Tools/spamfighter-exchange-module-3.5.0.0.zip/3667af4a9 [cited November 14, 2009]
45. SPAMfighter. Anti Spam Gateway for Mail Serves, http://www.spamfighter.com/Product_SMTP.asp [cited November 14, 2009]
46. GFI. Anti-spam filter for Exchange Server and Lotus Notes, http://www.gfi.com/mes/ [cited November 14, 2009];
47. SpamAssassin: Welcome to SpamAssassin, http://spamassassin.apache.org/ [cited November 14, 2009];
48. Stop spam with the Anti-Spam-SMTP-Proxy(ASSP), http://assp.sourceforge.net/ [cited November 14, 2009];
49. BullGuard. BullGuard Antivirus, Antispyware, Firewall, Spamfilter, Backup and Support, http://www.bullguard.com/main.aspx [cited November 14, 2009];
50. Cloudmark. Cloudmark Messaging Security - Block Spam, Fraud, Phishing & Viruses, http://www.cloudmark.com/en/home.html [cited November 14, 2009]
51. Spam Blocker from Clear My Mail, Spam Filter and Anti-Spam Solution, http://www.clearmymail.com/default.aspx [cited November 14, 2009]
52. Spam Blocker for Microsoft Outlook, http://www.mailfrontier.com/products_matador.html [cited November 14, 2009]
53. Spam Filter Express is anti-spam software and spam blocker to stop spam email, http://www.spam-filter-express.com/ [cited November 14, 2009]
54. Ferris Research. Industry Statistics (2009), http://www.ferris.com/?page_id=1078 [cited November 5, 2009]
55. Laboratory for Web Algorithmics, http://law.dsi.unimi.it/index.php?option=com_frontpage&Itemid=1 [cited November 25, 2009]
56. Girardi, C., Ricca, F., Tonella, P.: Web crawlers compared. International Journal of Web Information Systems Volume, 85–94 (2006)
57. Takemura, T., Ebara, H.: Economic loss caused by spam mail in japanese industries. RCSS Discussion Paper Series (2008)
58. Takemura, T., Ebara, H.: Spam mail reduces economic effects. In: Second International Conference on the Digital Society 2008(2008)
59. Morss, D., Harnett, D., Edwards, C.: State of Spam: A Monthly Report, Symantec (September 2009)
60. Ukai, Y., Takemura, T.: Spam mails impede economic growth. The Review of Socionetwork Strategies 1(1) (2007)

Spam 2.0: The Problem Ahead

Vidyasagar Potdar, Farida Ridzuan, Pedram Hayati, Alex Talevski,
Elham Afsari Yeganeh, Nazanin Firuzeh, and Saeed Sarencheh

Anti Spam Research Lab, Digital Ecosystems and Business Intelligence Institute
Curtin University of Technology, Australia
http://asrl.debii.curtin.edu.au
{v.potdar,a.talevski}@curtin.edu.au,
{farida.mohdridzuan,pedram.hayati}@postgrad.curtin.edu.au,
{yeganeh,sarenche,n_firoozeh}@iasbs.ac.ir

Abstract. Webspam is one of the most challenging problems faced by major search engines in the social computing arena. Spammers exploit weaknesses of major search engine algorithms to get their website in the top 10 search results, which results in higher traffic and increased revenue. The development of web applications where users can contribute content has also increased spam, since many web applications like blogging tools, CMS etc are vulnerable to spam. Spammers have developed targeted bots that can create accounts on such applications, add content and even leave comments automatically. In this paper we introduce the field of webspam, what it refers to, how spambots are designed and propagated, why webspam is becoming a big problem. We then experiment to show how spambots can be identified without using CAPTCHA. We aim to increase the general understanding of the webspam problem which will assist web developers, software engineers and web engineers.

Keywords: Webspam, CAPTCHA, Spambot, anti-spam, spambot navigation, Spam 2.0, Pligg spam.

1 Introduction

Spam in the context of email is defined as "*unsolicited, anonymous, commercial and mass email messages*". Spam originated via email and one of the first spam email dates back to early eighties, when a lawyer sent out an advertizing email on a newsgroup. Since then spam has evolved into what we know as spam today. A spammer is defined as "*an entity that is involved in spamming*". Spammers use many different mediums to spam web users, drifting from the traditional email approach to new approaches that are termed as *webspam or as we call it Spam 2.0 [31, 32].*

Webspam refers to the techniques employed by spammers to spread spam via websites in contrast to using emails [33]. Spammers now use blogs, forums, wikis [30, 34] or even develop their own websites to post advertizing material. Overall the motivation is still the same i.e. to generate revenue, increase page rank, promote product or services and steal user information [1].

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 400–411, 2010.

Spammers use a number of techniques to drive traffic to their websites and one of those is to fine tune their websites to deceive search engines in ranking their websites higher [29]. It is quite often seen that when you search for a particular keyword, you are taken to a website which does not relate to what you are looking for, but instead it is an advertizing page designed by spammer [36]. Such websites are carefully crafted to make the search engines believe that it is providing genuine content by implementing keyword stuffing, incorporating fresh content and several other strategies.

With sophisticated anti-spam techniques, it is now possible to get rid of the majority of spam; however a small percentage of spam can still escape these filters. Spammers rely on this small percentage of spam to attract their targets as they expect a portion of people to respond to their spam content. It is this response rate that keeps the spammers active [2]. Hence spammer aims to keep broadcasting very many spam messages on a regular basis as it increases their chances to find new targets. Spam is growing at a rapid pace and has become a big industry, mainly because it costs very little to send out millions of spam messages electronically [3]. According to [4], the cost to post an advertizing comment on a blog is very marginal, making spam campaigns extremely profitable particularly with favourable conversion rates.

This paper focuses on understanding how webspam operates and looks into the implications of webspam on productivity at work, consumption of network resources etc. We then study why spam is such a difficult problem to solve, where we look at technical, social and economic factors that affect spam. Later we outline the current solutions used to tackle spam and finally discuss our experimental results where we analyze spambot behavior.

1.1 How Does Webspam Operate?

Webspam operates by using a number of spambots. A *spambot* is a piece of code that is designed to post advertising comments on web applications like forums [35] & blogs. A collection of spambots forms a network that is referred as a *Botnet*, which is a network of infected machines that operate under the command of a *Botmaster*. Botnets normally use infected hosts to transmit spam [5]. Once the machine is infected it becomes a spam agent. Spammers have developed sophisticated web spamming tools that can automatically identify websites that host a particular type of web applications like blogs from Wordpress, forums from SMF, PHPBB etc.

Spambots can be categorized into *applications specific bots* that target web applications like Pligg, SMF, PHPBB, Wordpress etc and *websites specific bots* targeting websites with high traffic e.g. Amazon, CNET etc. Bots that target specific applications are customized so that can only be used to spam a subset of websites that are developed using a particular web application. These bots do their job perfectly and at times leave no trace of their activity or their origin. We will discuss later in our paper how the honeypot that we developed was able to capture these targeted spambots.

The harvesting activity i.e. finding out targets to spam is also done intelligently and to some extent the web applications are to be blamed. The simplest techniques to find whether a particular website is developed using a specific application is to look for unique text e.g. *“# Published News # Upcoming News # Submit a New Story”* is used in Pligg. This makes it easy to find targets and start spamming. The Botmaster usually does this job and sends individual bots to the selected sites for spamming.

On the other hand, website specific bots are extremely customized, usually developed for one particular website. The spammers study the structure of the website and develop an attack strategy to craft a bot that can bypass all the anti-spam mechanisms used. Even though these kinds of bots take a longer time to be develop they provide extremely good returns and are most difficult to detect. Hence understanding bot behaviour is becoming a key challenge for the research community and developing a system that can filter out bots from their behaviour is where the technology should be heading.

1.2 What Are the Implication of Webspam?

There have been several reports outlining the loss in labor productivity and network resources [6-10]. Although majority of these studies focused on email spam, this equally applies to webspam also. According to Nucleus research [8], in 2003 an average employee received 13.3 spam messages per day, which equated 6.5 minutes of their time to read and delete. This research also mentioned that for every 72 employees a company lost one employee to spam. In general, spam decreases individual's productivity directly/indirectly. The cost implications of spam show a very gloomy picture too.

- Nucleus research reported the average cost of spam per employee per year is $874
- [7] Estimated that spam is costing organizations $75 billion globally.
- [9] Reported that the labor loss caused by spam mail amounted to 21.6 billion dollars per year.
- [6] Reported that companies lost 20 billion dollars to buy additional servers in 2003 to manage spam.

Spam has implication on the global climate change too. According to a report by McAfee, the global energy consumption to process spam e-mail in 2008 was 33 billion kilowatt-hours (kWh), which can otherwise be used to support at least 2.4 million homes in a year [10]. Having understood the problem and its implications, we now look at why spam has become such a difficult problem to solve.

2 Why Is Spam a Difficult Problem to Solve?

The problem to address spam is not just technical in nature but it has economic and social dimensions too. Therefore, a spam management solution should incorporate technical, economic and social aspects too. In this section we highlight these issues in detail to understand why spamming is a difficult problem to solve.

2.1 Technical Dimension

Number of Bots
Internet is flooded by sheer number of bots that originate from numerous locations, with different technologies and strategies. Even though some of them can be easily

defeated, intelligent bots can recognize dead links, fake email addresses, identify spam traps etc making this a difficult task. Botnets are evolving rapidly because spammers continuously develop new techniques to hide bots [11], hence detecting bots is becoming harder. Moreover, the number of bots is increasing as well; about 30,000 new machines are infected and become bots every day [12]. According to Microsoft research the total number of bot-accounts signed up in hotmail in Jan 2008 is more than three times the number in Jun 2007 [11]. Here are some blacklists from different references showing the number of bots that have been detected as spammers [13-16].

Openness of the Internet

Openness of the Internet is a one key factor that allows spam to proliferate. Spammers can easily take this opportunity to manipulate this freedom since spammers have the same opportunity as other users in using any web application. So, if other users are allowed to write hundreds of comments everyday, spammers could do the same as long as their behavior matches that of a real human. Since no prior authorization is required to post a comment on a blog or forum, spammers can write advertizing comments on blogs without a problem. This inherent freedom allows spammers to take the risk as they have nothing to lose.

Gray Area between Spam and Non-spam

There is no distinct definition to differentiate real content from spam content, since there is no clear boundary between the two. One piece of information that is spam for one user may not be spam for the another, hence defining this boundary is a challenging task that has two major problems:

1. *False-positive*: it occurs when a legitimate content is flagged as spam.
2. *False-negative*: it occurs when spam content is flagged as genuine content.

The damage that can result from false-positives can be very serious and anti-spam solutions should be wary of this problem [18].

Vulnerable Web Applications

Many web applications are vulnerable to spam because these applications did not consider any anti-spam measures when they were built. Numerous web developers blindly use open source software to fast track development cycles; however they do not consider the impacts of adopting a web application without introducing any anti-spam measures. As a result, the number of vulnerable commercial or noncommercial websites has increased exponentially. The majority of these web applications provide features for users to contribute content, which is intelligently exploited by spammers. In addition, personal websites designed by novice developers are extremely vulnerable to spamming as well since they are not familiar with spam attack algorithms. Other than that there is no dedicated phase during the design and development stages of websites that caters for spam control.

Defensive Approach to Spam Management

Current spam management techniques take a defensive approach to address this problem. Many algorithms detect spam once it has entered the system. This approach is very dangerous in the case that spammers combine their spam content with a virus. Thus there needs to be a change in thinking on how spam is to be managed.

Intelligent Bots

Spambots have become extremely intelligent lately, they are not only capable of finding vulnerable websites and launch targeted attacks on but also devise real time strategies to get the highest return on their spam campaigns. Botnet's are widely spread across the web and largely operated by unsuspecting users' infected machines. Spam works in the favor of bots in three ways, *firstly* these machines are not blacklisted, so traffic originating from these sources is not blocked, *secondly* spammers do not need to pay for the resources and *thirdly* prosecuting the owners of infected machines is very difficult in the current legal system. Other than that, bot detection itself is a problem because of two reasons [19]: *firstly* bot's attacks are transient and *secondly* one specific bot does not send all the spam traffic.

Dynamically Adapting Spam Protection Schemes

Despite a lot of research focusing on developing better spam protection schemes, the nature of spam itself is changing rapidly to adapt to these new schemes. Spammers and the anti-spam companies are evolving simultaneously to adapt to the changing environment in the following ways;

⇒ spammers use keyword stuffing to fill their pages with specific keywords and anti-spam algorithms provide a list of the most used keywords which are being used by spammers.

⇒ spammers try to use unknown IP addresses for spamming and anti-spam software's provide blacklisted IP addresses from where spam originates. As a counter measure, spammers keep changing IP addresses or use proxy servers or infected machines to send spam.

⇒ spammers switch the nature of spam from text to graphic and anti-spam software's now use OCR to interpret textual content from images to detect spam.

Spammers are creative enough to create better content to prevent it from being detected easily by the anti-spam software's. Hence, while spam is changing dynamically, spam still remains an unsolved problem.

Trial and Error

Spambot can be easily created, in fact just a few lines of code is required to create a simple spambot. While the cost of creating spam is relatively cheap, not all spambots are good enough to breakthrough the anti-spam filters. Nevertheless, spamming is mostly based on trial and error. Given nearly infinite resources from infected machines, it is relatively easy for spammers to spam until the desired number of websites are infected within a given spam campaign.

2.2 Economic Dimension

Part Time Spamming Provides Good Income

Some people get hired to spam others and it is a very popular practice in developing countries like India [20], [22-24]. Spam jobs are advertised as email processing job that looks legal, however the actual idea is to get the spam comments or emails sent out in bulk. Most spamming jobs are advertized as home based work where an individual can work anytime that s/he wants and only an email account is required to do this job. We found some advertisement paying as low as 1 INR i.e. 2 cents per comment added to a website [20]. Since this job is performance driven, one gets paid more if one can spam more. This monetary benefit attracts a large number of people to the spamming business.

Cost to Spam Is Low

Another reason why spamming is difficult to control is that the cost to spam is very low. The majority of the cost is incurred by the infected machines. Unlike sending paper mail, the cost of sending spam is very minimal and the spammer ultimately profit even if a negligible rate of receivers respond to their advertisement [21].

2.3 Social Dimension

Human Spammers

As mentioned previously all bots may not be automated. Humans may be employed in developing countries to break strong CAPTCHA's which are used to limit the access of bots to sites. CAPTCHA generates textual or audio tests that are easy for human to solve, but difficult for bots [25].

Low Level of Awareness

It is also necessary to enhance people's awareness towards fighting spam. Spammer use general topics that interest wide range of the community such as cheap medicines, low priced air fares, low rate mortgages. In other words spammers abuse low level awareness of web users to spam e.g. spammer may suggest buying special product like medicines cheaper from a particular site and people believe it to be true and follow the links to a scam website operated by a spammer. Hence a general lack of awareness amongst the majority of web community is a key factor why spam is flourishing.

3 How Is Spam Managed Currently?

Currently spam is managed by three main approaches; these are Detection Approach, Prevention Approach & Attack Approach. All these approaches have had limited success so far and a reliable spam management strategy should include all the three approaches to achieve maximum spam protection. We now explain each of these approaches in detail.

3.1 Detection Approach: Spam Content Is Identified and Filtered Out

This approach is one of the first approaches developed and implemented to manage spam. The basic idea here is to identify and filter out spam content from genuine content. To achieve this, several techniques have been proposed in the literature and many of them are currently implemented in commercial anti-spam toolkits. For example, detection methods can be categorized into two: content based & metadata based. The former uses content to analyze spam and hence is computationally intensive and more reliable whereas the latter only uses metadata i.e. links or url or email headers and is relatively fast but comparatively less reliable. Both approaches rely on data mining techniques which can either be supervised, semi-supervised or unsupervised. Supervised methods require a labeled data set for spam classification whereas unsupervised do not [26]. Some anti-spam methods are language dependent and hence may not be able to apply to non English language spam, which can be a problem.

3.2 Detection Approach: Users Flag Content as Spam

This approach is a subset of the detection strategy, where the end users are involved in helping to fight spam. This feature is commonly seen in free email services like Yahoo Mail, Hotmail or Gmail etc., where the users have the option to select an email and tag it as Spam. Lately this feature has also become popular in blogs and forums. This feature is very good, as it can help the anti-spam detection algorithms to build up a spam data set; however, the downside of this approach is that spammers can equally use this feature to tag genuine content as spam. So studying the effectiveness of this method is very interesting. Currently there are no publicly available results to show whether this strategy is working [27-28].

3.3 Prevention Approach: CAPTCHA

Prevention approach was developed to defeat spambots by requesting spambots to go through an online test named as CAPTCHA. The aim of this test was to distinguish human users from spambots. The test requires the user to type in unclear, curvy or ghostly characters from in an image to a registration form. Most users should be able to surpass this test easily but bots would fail even if they use optical character recognition techniques. CAPTCHA is used in almost all commercial emails sites (Yahoo, Google), online forum, blogs, and social networking sites to prevent automated registrations. CAPTCHA also helps bloggers in dealing with comment spam. However there are some drawbacks e.g.:

⇒ it relies on human visibility, hence it is inconvenient for users with bad vision.

⇒ at times it is even very difficult for normal users to decipher the CAPTCHA.

⇒ free CAPTCHA servers incur longer delays in processing

⇒ many spammers have developed OCR techniques to automatically read CAPTCHA.

⇒ as Optical Character Recognition (OCR) techniques improve, CAPTCHA’s images become harder and hard to decipher even by humans. This damages the typical users web experience

⇒ as computers get more powerful, they will be able to decipher image and voice CAPTCHA requests similar to humans.

3.4 Attack Approach: Poisoning Spammers Database

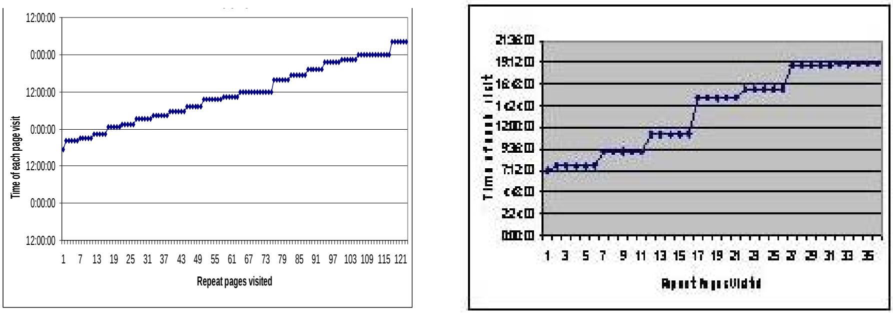
This is a relatively new approach to address spam. The basic idea is to infiltrate spammers’ database and poison it with fake email address, with an aim to reduce the effectiveness of spamming campaigns. So instead of waiting for the spammers to attack, this method takes an active approach towards attacking spammers. This method generates random email addresses and waits for bots to index them. Once the spammer realizes that their database is full of invalid information, it will reduce the effectiveness of their spam campaigns. However the main concern is that whether the list of random emails generated are really invalid or they do belong to someone. An additional concern is that spammers may not bother if their database is poisoned, they may just send spam to all the emails addresses they have, since they are not using their resources any way. WPOISON and SUGARPLUM are examples of such services available on the Internet.

4 Experimental Results

In order to monitor & analyze the spambot behavior we conducted an experiment by setting up a honeypot which had public access to the Internet. We decided to use Pligg as a web application to assess the spambot behavior because Pligg's anti-spam features are very weak. We modified Pligg source code to integrate it with our user navigation tracking (UNT) functionality to track spambot navigation behavior. We tracked: *url visited, session identification, user agent, referral link, start time, last login time, total active time, total time per visit, average time user spends on each session and total active time*. To advertize the honeypot we listed the URL of this honeypot on several sites and we got sufficient spambots. We now explain our 4 main observations from this experiment.

4.1 Observation 1: Spambots Failed Attempts

Over a period from 16th May 2009 to 9th July 2009 we got 412 unique spambot visiting our site. Overall there were 599 visits from different bots i.e. repeat bots. From the total registered users 9 spamuser's attempted to add content on this site. Fig. 1(a) & 1(b) shows failed attempts by two spambots to add comments to Pligg Story Page, since we had changed the HTML code to trap the bot to study its navigation



(a) Bot 1 attempted 21 times to add a comment (b) Bot 2 attempted 7 times to add a comment

Fig. 1. Spambots failed attempts to add comments to Pligg Story Page

patterns. We found that spambots were programmed to follow a standard path when attempting to add content and that was as follows: Homepage →Show Story →Show Story →Show Story →Show Story.

4.2 Observation 2: Navigation Pattern Detection

When we activated content submission some other spambots started submitting content and had the following navigation pattern User Page → Submit News → Submit News → Submit News → Submit News → Upcoming News. Fig. 2 shows the similarity between navigation behaviors of two spambots. We also created an interface to analyze each record to help us further discover spambots behavior.

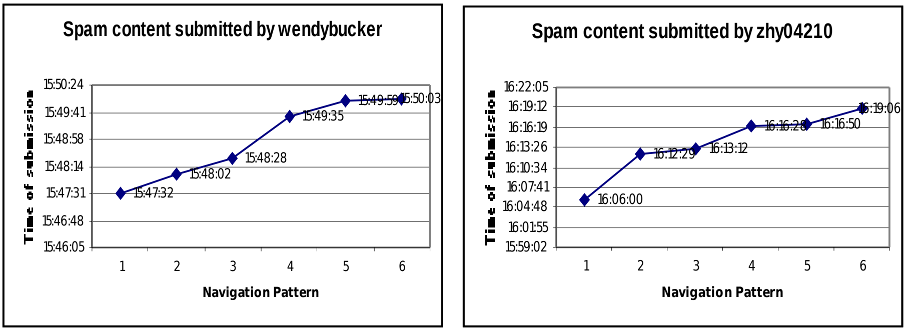


Fig. 2. Bot 3 & Bot 4 submitting spam on the Pligg website following similar pattern

4.3 Observation 3: Normal Navigation Pattern

Some spambots used 8 different user agents or modified the header info i.e. browsers and operating systems to access the Pligg website. The normal navigation pattern for almost all users was as follows: Register →Register →Submit →Submit →Upcoming News.

4.4 Observation 4: Motivation of Navigation Pattern

We assume this pattern of navigation shows that the spambot first wants to ensure that their account has been created, then they want to ensure that their content has been submitted and finally checking whether that content is live or not. If the content is live they consider their job to be complete and do not visit the website for a predefined interval of time.

5 Discussion and Conclusion

We have seen that the current anti-spam filtering techniques are not effective in combating webspam. Purely on a technical front addressing spam problem is going to be a difficult challenge. We have reached a stage where the technical battle between spammers and anti-spam providers has reached a tipping point. We need to look at how we can solve the spam problem by relying on non-technical measures e.g. increasing awareness amongst the community about spam could be a good alternative. However the effectiveness of such an idea needs to be investigated. Australian media and communication authority has a website on spam, scam and fraud, but how many people actually visit this site is a different question. We believe people would visit the website once they become aware that they have been stung by a spammer, by then it is too late. Alternatively should we be looking at some management or legal solutions to combat spam? May be introducing tough money laundering legislation may be a good choice however the problem arises when spammers (or spam servers) are physically located in a different country where there is no spam legislation. At this moment, spammers are exploiting these legislative loop holes. Other than this, would it be possible to change the monetary equation such that spamming becomes an unattractive job? If some strategies can be investigated along these lines it would be very promising. The issue becomes even worse if money is not in the equation, some research indicates that spammers may be politically or even religiously motivated, in which case their motivations may be similar to computer virus creators. So the question now lies whether spam will infiltrate other communication mediums such as Digital TVs? Radios? Wireless sensor networks?

References

1. Hayati, P., Potdar, V.: Evaluation of spam detection and prevention frameworks for email and image spam: a state of art. In: Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services, ACM, Linz (2008)
2. Aaron, W.: Ending spam's free ride. netWorker 7(2), 18–24 (2003)
3. Fazlollahi, B.: Strategies for Ecommerce Success, p. 300. IGI Publishing (2002)
4. Chris, K., et al.: Spamalytics: an empirical analysis of spam marketing onversion. In: Proceedings of the 15th ACM conference on Computer and communications security, Alexandria, Virginia, USA, CM (2008)
5. Moheeb Abu, R., et al.: A multifaceted approach to understanding the botnet phenomenon. In: Proceedings of the 6th ACM SIGCOMM conference on Internetmeasurement, ACM, Rio de Janeiro (2006)
6. Group, e., Spam: By Numbers (June 2003)
7. Neal, L.: Vendors Fight Spam's Sudden Rise. Computer 40(3), 16–19 (2007)
8. Nucleus, R.: Spam: The silent ROI Killer. Research Note D59 (2003), 14 July 2009 [cited; Available from: http://www.nucleusresearch.com
9. Rockbridge, A.I.: National Technology Readiness Survey: Summary Report 2005 (2004)
10. Vrhnjak, S., Staff, C.: Spam is a big polluter in more ways than one (2009)
11. Yao, Z., et al.: BotGraph: large scale spamming botnet detection. In: Proceedings of the 6th USENIX symposium on Networked systems design and implementation, USENIX Association, Boston (2009)
12. Husna, H., et al.: Behavior Analysis of Spambotnets. In: 3rd International Conference on Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008, Bangalore, pp. 246–253 (2008)
13. [cited 13] (July 2009), Available from: http://www.joewein.net/dl/bl/from-bl.txt

14. Antispam. [cited 13] (July 2009) Available from: `http://antispam.imp.ch/swinoguri-rbl.txt`
15. Joewein. [cited 13 July 2009] Available from: `http://www.joewein.net/dl/bl/dom-bl.txt`
16. Juniper. [cited 13 July 2009]; Available from: `http://www.juniper.net/security/spam/`
17. Lowd, D., Meek, C.: Good Word Attacks on Statistical Spam Filters. In: Second Conference on Email and Anti-Spam (CEAS), Palo Alto, CA (2005)
18. Cunningham, P., et al.: A Case-Based Approach to Spam Filtering that Can Track Concept Drift. In: Ashley, K.D., Bridge, D.G. (eds.) ICCBR 2003. LNCS, vol. 2689, p. 3. Springer, Heidelberg (2003)
19. Yinglian, X., et al.: Spamming botnets: signatures and characteristics. In: Proceedings of the ACM SIGCOMM 2008 conference on Data communication, ACM, Seattle (2008)
20. Workathome. [cited 13 July 2009]; Available from: `http://www.workathomeforum.in/online-adplacing-homejob.htm`
21. Leiba, B., Borenstein, N.: A multifaceted approach to spam reduction. In: First Conference on Email and Anti-Spam, CEAS (2004)
22. Cobb, S.: The Economics of Spam (2003)
23. Rich, L.L.: Internet Legal Issues: SPAM (1999)
24. Schwartz, E.I.: Spam Wars (2003)
25. Halprin, R.: Dependent CAPTCHAs: Preventing the Relay Attack, 26 (2009)
26. Hayati, P., Potdar, V.: Toward Spam 2.0: An Evaluation of Web 2.0 Anti-Spam Methods. In: 7th IEEE International Conference on Industrial Informatics (INDIN 2009), Cardiff, Wales, ‘ (2009)
27. Sheffield, M.: ’Flag Spam,’ the Preferred Tool of the Left’s Web Censors. 2008 [cited 14,]; 2008/10/07/flag-spam latest-tool-censors-left (July 2009), Available from: `http://newsbusters.org/blogs/matthewsheffield/`
28. userscripts.org. Flagging Content Feature. [cited 14 July 2009]; Available from: `http://userscripts.org/topics/1362`
29. Hayati, P., Potdar, V.: Spammer and Hacker, Two Old Friends. In: 3rd IEEE International Conference on Digital Ecosystems and Technologies (DEST 2009), Istanbul, Turkey (2009)
30. Hayati, P., Potdar, V.: Toward Spam 2.0: An Evaluation of Web 2.0 Anti-Spam Methods. In: 7th IEEE International Conference on Industrial Informatics (INDIN 2009), Cardiff, Wales (2009)
31. Hayati, P., Chai, K., Potdar, V., Talevski, A.: Behaviour-based web spambot detection by using Action Time and Action Frequency. In: The 2010 International Conference on Computational Science and Applications, Springer, Heidelberg (2010)
32. Hayati, P., Potdar, V., Chai, K., Talevski, A.: Web Spambot Detection Based on Web Usage Behavior. In: The International Conference on Advanced Information Networking and Applications, AINA 2010 (2010)
33. Ridzuan, F.H., Potdar, V., Talevski, A.: Key Parameters in Identifying Cost of Email Spam. In: The 2010 International Conference on Computational Science and Applications, Springer, Heidelberg (2010)
34. Ridzuan, F.H., Potdar, V., Talevski, A.: Key Parameters in Identifying Cost of Spam 2.0. In: 24th IEEE International Conference on Advanced Information Networking and Applications, AINA 2010 (2010)

35. Sarencheh, S., Potdar, V., Yeganeh, E.A., Firouzeh, N.: Semi-Automatic Information Extraction from Discussion Boards with Applications for Anti-Spam Technology. In: International Conference on Computational Science & its Applications (ICCSA 2010), Springer, Heidelberg (2010)
36. Hayati, P., Chai, K., Potdar, V., Talevski, A.: HoneySpam 2.0: Profiling Web Spambot Behaviour. In: Yang, J.-J., Yokoo, M., Ito, T., Jin, Z., Scerri, P. (eds.) PRIMA 2009. LNCS, vol. 5925, pp. 335–344. Springer, Heidelberg (2009)

Tag Recommendation for Flickr Using Web Browsing Behavior

Taiki Takashita, Tsuyoshi Itokawa, Teruaki Kitasuka, and Masayoshi Aritsugi

Computer Science and Electrical Engineering
Graduate School of Science and Technology
Kumamoto University, Kumamoto 860-8555, Japan
{reo@dbms., itokawa@, kitasuka@, aritsugi@}cs.kumamoto-u.ac.jp

Abstract. It is fun to share photos with other people easily and effectively. For it, a tag recommendation system for Flickr is developed in this paper. We assume that there are some relations between a photo of which a user attempts to update tags and webpages that the user has browsed. In our system, Web browsing behavior of a user is exploited to suggest not only candidate tags to be added to but also candidate tags to be deleted from a photo in Flickr to the user. We discuss how to implement the system in this paper. We also report some experimental results to show the effectiveness of our system.

1 Introduction

Since the price of digital cameras with high functions is becoming lower and lower, a large number of photos are taken and shared among people in the Internet through online photo services such as Flickr [4]. Although many researchers have been attracted to content based retrievals so far, a feasible way to handle the photos effectively is to add appropriate tags to each photo according to its contents. Not to mention, the more appropriate tags are added to photos, the more effectively they can be shared. Folksonomy [10] is a framework for letting each photo have good tags as its metadata. In fact, Flickr has a function of Folksonomy; users can upload photos and update their tags at any time.

However, the quality of tags added to a photo in Flickr may not be high. Observing general tag characteristics in Flickr [13], the number of tags that 64% of the photos have is below three. This may be because selecting appropriate words and adding them to photos is a troublesome process for users. On the other hand, some photos in Flickr have irrelevant or inaccurate tags, referred to as meta noise [11]. These characteristics of tags will make the accuracy of photo retrievals worse naturally.

In this paper, we develop a tag recommendation system for Flickr. Our system can suggest not only candidate tags to be added to but also candidate tags to be deleted from a photo in Flickr. We focus on user behavior that most Flickr users browse webpages, and assume that there are some relations between a photo of which a user is going to update tags and webpages that the user has browsed, because the user must be interested in both. We attempt to exploit the relations

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 412–421, 2010.

in tag recommendation. Our approach is based on the idea we have studied in [15,14], in which user preference is extracted from Web browsing behavior and exploited in spam mail filtering. One of the merits using Web browsing behavior is that we can collect necessary information for generating tag recommendation from users without their special efforts.

The rest of this paper is organized as follows. Section 2 mentions related work and compares with our work. Section 3 proposes a tag recommendation system using Web browsing behavior. Section 4 reports some experimental results to evaluate our proposal, and Section 5 concludes this paper.

2 Related Work

There have been studies on image annotation [7,8,3,9] in which a new image data is added tags derived from those already added to stored images. In [7,8,3], similarity between the new image and images already stored is calculated and tags that similar images have are used to annotate the new image. This idea works well when we can suppose that stored images have appropriate tags. The dataset come from Corel Stock Photo cds, in which each image is assigned an appropriate annotation of 1-5 keywords, are often used to evaluate their studies. On the other hand, we cannot suppose that photos in Flickr have appropriate tags, and thus we need another approach.

[9] proposes a dual cross-media relevance model for automatic image annotation, which attempts to estimate the joint probability of images and words, and evaluates it with Corel and Web datasets. They concludes that the model is potentially applicable for the Web image annotation, the performance seems not so good, though.

Similar to our work, [13,6] study tag recommendation for Flickr based on user interaction, that is, photos and their tags of a user. In [13], tag co-occurrence for photos is calculated using tags appearing both in a user's tagging history and in Flickr, and used to generate recommended tags to be added. In [6], tag recommendation is obtained using both a Naive Bayes classifier on a user's tagging history and tf-idf based global information. In their schemes, the quality of tag recommendation results of depends on information about tags in the user's tagging history.

In contrast, we attempt to exploit Web browsing behavior of a user in tag recommendation for Flickr. We have studied Web browsing behavior for spam filtering [15,14], which were inspired by [1]. In conventional spam filtering, the maintenance of filtering methods to adapt changes with time is generally a tedious and expensive task. To reduce the troublesome maintenance, we exploit Web browsing behavior to collect necessary information for the maintenance from webpages that users browsed [15,14]. In this paper, we apply the idea to tag recommendation for Flickr. Since most Flickr users are supposed to browse webpages regularly, we can collect necessary information for tag recommendation without users' special efforts. In addition, we think the amount of information from Web browsing behavior tends to be larger than that from tagging history, thereby generating better tag recommendation with using Web browsing behavior.

3 Tag Recommendation Using Web Browsing Behavior

Our system introduced in this paper generates tag recommendation with using words extracted from webpages that a user browsed. Given a photo and its current tags, our system calculates tag co-occurrence using the extracted words and generates candidate tags both to be added to the photo and to be deleted from the current tags. Our tag recommendation consists of two procedures, namely, words extraction from Web browsing behavior of a user and tag recommendation generation. Hierarchical clustering [12] is exploited for reducing costs for handling necessary information including tags, words, and webpages in our system, as shown in the following. We also discuss tag recommendation re-generation with using recommended tags generated in our system.

3.1 Words Extraction from Web Browsing Behavior

When a user browses webpages day by day, our system collects the URLs and their HTML sources of the pages to extract words or phrases of interest. We suppose webpages that a user browsed are interesting to the user. For simplicity, we do not take the degree of interest of each page into account in the current implementation. Thus, our system collects a webpage once per day even if a user browsed the page many times, thereby reducing the cost for this collection.

In practice, we need to consider careful handling in collecting Web browsing behavior. For example, we may need to exclude such webpages that a user browsed accidentally from the collection. We think it would be helpful to update the information of URLs and HTML sources periodically; we do not go through this in this paper.

The collected HTML sources are analyzed morphologically, resulting in words or phrases. Each of the extracted words or phrases consists of 1-3 nouns or an adjective and 1-2 nouns in the current implementation. On the other hand, we collect information about tags in Flickr through Flickr API [5]. We exclude such words or phrases that appear as tags in Flickr less than N_{wf} times, which can be processed using the collected information, from the extraction.

We next calculate tf-idf scores of the collected words or phrases by means of web services offered by a company like Yahoo! and Google. We store words or phrases having high tf-idf scores in each webpage as characteristic words or phrases of the webpage. We express the number of words or phrases to be stored per a webpage as N_w in this paper.

Similar to [12], we then process hierarchical clustering with group average method on the collected webpages using their corresponding vectors, each element of which is the tf-idf score calculated above, in order to reduce the cost for generating tag recommendation, which will be described in the next subsection. Each webpage is initially treated as a cluster, and the hierarchical clustering process performs until the similarity between any of two clusters is less than Th_s. After clustering on webpages, we process hierarchical clustering on words or phrases in every generated webpage cluster. A feature vector for a word or phrase consists of relevancies to the other words or phrases. The hierarchical clustering process on words or phrases performs until the number of word or phrase clusters is less than N_{cw}. In the current implementation, we use the concept of Normalized Google Distance (NGD) [2] for the relevancies. Given two words or phrases v and w, the NGD is defined as follows:

$$NGD(v,w) = \frac{\max\{\log f(v), \log f(w)\} - \log f(v,w)}{\log G - \min\{\log f(v), \log f(w)\}}, \tag{1}$$

where $f(v)$ is the number of retrieval results using the search term v, $f(v,w)$ is the number of retrieval results using the search terms v and w, and G is the number of webpages indexed by Google. To use it for this study, we set $f(v)$ to the number of retrieved photos using the search term v from Flickr, and G to 40 million, which was estimated from the number of photos having at least two tags reported in [13]. Then, similar to [9], the relevancy score of two words or phrases v and w in this study is calculated as follows:

$$Relevancy(v,w) = \exp[-\alpha \cdot NGD(v,w)], \tag{2}$$

where α is a parameter.

3.2 Tag Recommendation Generation

Given a photo and its tags in Flickr, we generate a recommendation to the photo, by retrieving relevant words or phrases to the tags from those extracted in the previous procedure.

When the number of tags is larger than N_{ct}, we first process hierarchical clustering on the tags to make the number of clusters be less than or equal to N_{ct}. The tag that appears in Flickr most times among those in each tag cluster is called as the representative tag of the tag cluster. Then, we calculate relevancy scores between every pair of a representative tag and a webpage cluster. A relevancy score of them is mean value of relevancy scores between a representative tag and N_w words or phrases in a webpage in a webpage cluster. We consider webpage clusters having more than or equal to Th_r relevancy score to a representative tag as those relevant to the representative tag's tag cluster. We next find from the characteristic words in a webpage such a keyword in a webpage cluster that has the largest relevancy score to tags in relevant tag clusters to the webpage cluster. We then find a word or phrase cluster including a keyword and its neighbor two clusters in cluster hierarchy, and calculate mean value of relevancy scores between a word or phrase in the three clusters and every tag in a tag cluster. We select words or phrases having high mean value of the relevancy scores. We express the number of selected words or phrases as N_r in this paper.

By performing these processes, we finally obtain words or phrases that are relevant to each tag cluster from webpages that the user browsed. Here we can say that the number of words or phrases relevant to each tag cluster indicates the relevancy between the tag cluster and the given photo, i.e., if the number is large the tag must be appropriate to the photo. Also, we can consider tags that have a small number of relevant words or phrases should be deleted.

In generating a list of ranked recommended words or phrases to be added, we consider in this paper three methods for scoring words or phrases: one is to use the relevancy between a recommended word or phrase and tags, another is to use the relevancy and Inverse Document Frequencies (IDF), and the other is to use the relevancy and modified IDF. We call them as Relevancy-only, Relevancy-with-IDF, and Relevancy-with-modified-IDF, respectively, in this paper.

A score in Relevancy-only is defined as follows:

$$Score_{Rel-only}(w \in W) = \frac{1}{|Q|} \sum_{q \in Q} Relevancy(w, q), \tag{3}$$

where W is the set of all words or phrases in the list and Q is the set of tags that have already been added to the given photo.

In Relevancy-with-IDF, we first calculate scores using (3) and have a sorted list consisting of N_o words or phrases. Then, we re-score them using IDF as follows:

$$Score_{Rel-with-IDF}(w \in W) = \{\frac{1}{|Q|} \sum_{q \in Q} Relevancy(w, q)\} / \log(\frac{m}{m^{(w)}}), \tag{4}$$

where m is the number of photos in Flickr and $m^{(w)}$ is the number of photos having tag w. By using the score that is weighted with IDF as in (4), words that often appear in Flickr can have relatively high score.

Note that it must be useless to suggest too general or too specific tags in tag recommendation. We thus think another scoring method called Relevancy-with-modified-IDF. As in the above, we first calculate scores using (3) and have a sorted list consisting of N_o words or phrases. Then, we re-score them using the formula defined as follows:

$$Score_{Rel-with-mod-IDF}(w \in W) = \{\frac{1}{|Q|} \sum_{q \in Q} Relevancy(w, q)\} / \log(\log(\frac{m}{m^{(w)}})). \tag{5}$$

Our system finally provides to the user the list of ranked words or phrases as candidate tags to be added.

Our system can also suggest tags to be deleted from the given photo. If a tag has a small number of webpage clusters that are relevant to the tag, we can consider the tag to be uninteresting to the user and thus suggest them to be deleted to the user. In the current implementation, tags having the smallest number of webpage clusters that are relevant to the tags are suggested to be deleted.

3.3 Recommendation Re-generation with the Tag Recommendation

According to the tag recommendation generated above, the user can add appropriate tags to and delete inappropriate tags from the photo. Obviously, the quality of the resultant tags is expected to become better than that of the previous tags. Our system can generate another tag recommendation by using the better tags. In generating tag recommendation of this round, we exclude the words or phrases which our system has already suggested in the previous round. Thus, if most appropriate tags have already been recommended in the previous round, the accuracy of the results in this round may become relatively low compared with those generated in the previous round.

4 Experimental Evaluation

4.1 Environment

We empirically evaluated our system with real data in Flickr. The data consisted of three sets of 100 photos and their tags extracted from Flickr using search terms “mlb”, “nba”, and “nfl”. The numbers of tags and distinct tags were 5270 and 2149, respectively. To model Web browsing behavior, we collected webpages consisting of three sets of 100 webpages extracted from Yahoo! using the search terms and 700 webpages randomly chosen from other categories than the three terms in Yahoo!. The number of distinct words or phrases collected from the webpages were 4237. TreeTagger [16] was used in morphological analysis. We used WebSearch API offered by Yahoo! [17] in calculating tf-idf scores.

Table 1 shows the settings of parameters, which we mentioned in the previous section, used in the experiments. The values were set according to results of our preliminary experiments. Parameters N_* were decided mainly by the limitation of the computing power of the machine we used in the experiments. On the other hand, we may need to study a good way to decide parameters Th_* for performance. For example, it might be better to take into account the data and/or their density distribution in deciding the parameters; this will be included in our future work.

Table 1. Parameter settings

N_{wf}	N_w	Th_s	N_{cw}	α	N_{ct}	Th_r	N_r	N_o
100	20	0.8	5	1	5	0.5	50	20

As in [13] and others, we used Mean Reciprocal Rank (MRR), Success at rank k (S@k), and Precision at rank k (P@k) as metrics for evaluating recommended tags to be added in this study. Recommended tags to be deleted were evaluated in terms of precision. Because the data we used in the experiments come from Flickr and thus there were no grand truth on the data, all the results reported in the following were assessed whether they were good or not by the authors manually.

4.2 Results

Table 2 shows the results of recommended tags to be added obtained from the first round of recommendation generation. Since the used data in the experiments come from Flickr, they must be different from those used in the other related studies [9,13,6]. Although it may thus be meaningless to compare the results with them because of the difference, we can see that the results were almost better. As a result, we consider that the performance of our system is not bad, and that using Web browsing behavior is effective in generating tag recommendation for Flickr.

Table 2. Results on recommended tags to be added

	MRR	S@5	S@10	P@1	P@5	P@10
Relevancy-only	0.66	0.87	0.93	0.52	0.41	0.32
Relevancy-with-IDF	0.74	0.91	0.96	0.63	0.46	0.35
Relevancy-with-modified-IDF	0.79	0.94	0.96	0.68	0.49	0.36

As shown in the table, Relevancy-with-modified-IDF could have the best performance among the three, and Relevancy-with-IDF outperformed Relevancy-only, in terms of all aspects, especially in terms of precisions at rank k. According to the results, we can say that the scoring method with modified IDF works well to sort candidate tags.

Figure 1 shows a sample photo and top 10 tags recommendations to be added in the three methods. Recommended tags with mark "$\surd$" were assessed as good in the experiments. As shown in the figure, Relevancy-with-modified-IDF can give good tags better scores than the other two methods. In other words, good words recommended in both Relevancy-only and Relevancy-with-IDF are highly scored in Relevancy-with-modified-IDF.

Relevancy-only	Rel-with-IDF	Rel-with-mod-IDF
rays	$\surd$ sports	$\surd$ sports
yankees	york	$\surd$ stadium
$\surd$ diamond	$\surd$ game	$\surd$ game
pitcher	bay	rays
$\surd$ stadium	$\surd$ team	$\surd$ team
reds	$\surd$ stadium	york
$\surd$ league	pittsburgh	$\surd$ league
tigers	$\surd$ league	yankees
giants	monument	bay
$\surd$ sports	rays	$\surd$ diamond

Fig. 1. A sample photo and tag recommendations to be added

Table 3 shows the results on recommended tags to be deleted. 927 tags were considered to be deleted by our system for the 300 photos. That is, three tags for one photo were averagely recommended to be deleted, and more than a half of them were assessed as good. Note that the tags having the smallest number of webpage clusters are suggested to be deleted in our current implementation as described before. If we adjusted this strategy appropriately, we could improve the results on recommended tags to be deleted; this will be included in our future work. Note also that, to our knowledge, there has been no other work that attempts to suggest tags to be deleted.

Table 3. Results on recommended tags to be deleted

# of results	# of good	precision
927	523	0.56

Figure 2 shows a sample photo and tag recommendation to be deleted. This was a good situation because all the tags were assessed as good in the experiments.

√ msh
√ monthlyscavengerhunt
√ msh0306

Tag recommendation to be deleted

Fig. 2. A sample photo and tag recommendations to be deleted

Table 4 shows the results on recommended tags to be added generated using tag recommendation results of the first round. The results of the first round consisted of 10 tags to be added and tags to be deleted. We excluded the tags in generating another tag recommendation in this second round of processing. The results were worse than those shown in Table 2. This is mainly because many good tags have already been suggested in the first round. Similar to the results of the first round of processing, Relevancy-with-modified-IDF outperformed the other two methods in this round, too.

Figure 3 shows tag recommendation results for the sample photo shown in Fig. 1. We can see in the figure that good words that have not been recommended in the first round can be suggested in this round.

From the results shown in Tables 2 and 4 and in Figs. 1 and 3, we can say that the tag recommendation with recommended tags can suggest good tags by using the results of the previous round.

Table 4. Results on recommended tags to be added generated by twice recommendation processes

	MRR	S@5	S@10	P@1	P@5	P@10
Relevancy-only	0.43	0.61	0.72	0.30	0.23	0.18
Relevancy-with-IDF	0.48	0.71	0.83	0.32	0.26	0.19
Relevancy-with-modified-IDF	0.53	0.74	0.85	0.38	0.29	0.20

Relevancy-only	Rel-with-IDF	Rel-with-mod-IDF
mariners	√ ball	√ fans
sox	championship	series
√ team	yankees	tigers
√ players	√ fans	pirates
angels	tigers	√ players
√ fans	pirates	sox
mets	series	angels
pirates	angels	mariners
nationals	nationals	nationals
phillies	mets	pitcher

Fig. 3. Tag recommendations to be added in the second round

5 Conclusion

In this paper, we have proposed a tag recommendation system for Flickr using Web browsing behavior. Our system can suggest not only tags to be added but those to be deleted as well. It should be noted that we do not need users' special efforts in generating tag recommendation in our system, since most Flickr users are supposed to browse webpages regularly and our system exploits the Web browsing behavior for tag recommendation. According to the results of our experimental evaluation, our system can generate tag recommendation effectively.

In our system, many parameters should be set appropriately. How to decide them will be included in our future work. The essential idea to exploit Web browsing behavior is different from that of existing related work including [13,6]. Thus, it is interesting to combine with them for getting better performance.

References

1. Budzik, J., Hammond, K.J.: User interactions with everyday applications as context for just-in-time information access. In: IUI 2000. In: Proceedings of the 5th international conference on Intelligent user interfaces, pp. 44–51. ACM, New York (2000)
2. Cilibrasi, R., Vitanyi, P.: Automatic extraction of meaning from the web. In: Proc. IEEE International Symposium on Information Theory, July 2006, pp. 2309–2313 (2006)

3. Feng, S.L., Manmatha, R., Lavrenko, V.: Multiple bernoulli relevance models for image and video annotation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1002–1009. IEEE Computer Society Press, Los Alamitos (2004)
4. Flickr: http://www.flickr.com/
5. Flickr API: http://www.flickr.com/services/api/
6. Garg, N., Weber, I.: Personalized, interactive tag recommendation for flickr. In: RecSys 2008: Proceedings of the 2008 ACM conference on Recommender systems, pp. 67–74. ACM, New York (2008)
7. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: SIGIR 2003: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 119–126. ACM, New York (2003)
8. Lavrenko, V., Manmatha, R., Jeon, J.: A model for learning the semantics of pictures. In: Neural Information Processing Systems, NIPS (2003), http://books.nips.cc/papers/files/nips16/NIPS2003_AA70.pdf
9. Liu, J., Wang, B., Li, M., Li, Z., Ma, W., Lu, H., Ma, S.: Dual cross-media relevance model for image annotation. In: MULTIMEDIA 2007: Proceedings of the 15th international conference on Multimedia, pp. 605–614. ACM, New York (2007)
10. Mathes, A.: Folksonomies – cooperative classification and communication through shared metadata. Tech. rep., Computer Mediated Communication (LIS590CMC), University of Illinois, Urbana-Champaign (December 2004), http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html
11. Peterson, E.: Beneath the metadata – some philosophical problems with folksonomy. D-Lib. Magazine 12(11) (November 2006), http://dx.doi.org/10.1045/november2006-peterson
12. Shepitsen, A., Gemmell, J., Mobasher, B., Burke, R.: Personalized recommendation in social tagging systems using hierarchical clustering. In: RecSys 2008: Proceedings of the 2008 ACM conference on Recommender systems, pp. 259–266. ACM Press, New York (2008)
13. Sigurbjörnsson, B., van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: WWW 2008: Proceeding of the 17th international conference on World Wide Web, pp. 327–336. ACM Press, New York (2008)
14. Takashita, T., Itokawa, T., Kitasuka, T., Aritsugi, M.: Extracting user preference from Web browsing behaviour for spam filtering. Int. J. Adv. Intell. Paradigms 1(2), 126–138 (2008)
15. Takashita, T., Itokawa, T., Kitasuka, T., Aritsugi, M.: A spam filtering method learning from Web browsing behavior. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part II. LNCS (LNAI), vol. 5178, pp. 774–781. Springer, Heidelberg (2008)
16. TreeTagger:
http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
17. Yahoo! Search Web Services: http://developer.yahoo.com/search/

USABAGILE_Web: A Web Agile Usability Approach for Web Site Design

Gladys Benigni[1,2], Osvaldo Gervasi[1], Francesco Luca Passeri[1], and Tai-Hoon Kim[3]

[1] Department of Mathematics and Computer Science, University of Perugia,
via Vanvitelli, 1, I-06123 Perugia, Italy
gbenigni@gmail.com, osvaldo@unipg.it

[2] Dept. Informatica, Isla de Margarita
Universidad de Oriente, Nucleo Nueva Esparta, Venezuela

[3] Science and Engineering Research Support Center
Hannam University
Ojeong-dong Daedeok-gu
Daejeon306-791, Korea
taihoonn@empal.com

Abstract. The present research presents an agile methodology designed for the Web, to design or to re-engineer a software product by analyzing the product's architecture, developing new user interface's prototypes, and testing them using detailed usability criteria. The proposed methodology represents an innovative approach to provide to the end-users a product easily comprehensive and easy to use, based on modern user-machine interfaces. For this reason the approach represents an important achievement towards the comprehension of the inner mechanisms on which is based the human-computer interaction.

1 Introduction

We propose a method for the design of accessible web sites based on the agile usability approach. The present method is particularly designed for implementing or restyling web sites requiring the evaluation of its usability. However the proposed method can be applied also to standalone software interfaces. The agile methodology [1] has been adopted both for developing the interface and for the evaluation of its usability.

The method, named *USABAGILE_Web*, is an iterative software engineering approach, acting on single interfaces like web pages or standalone applications. In each working unit of the developing process, we perform the analysis of the usability of the interface, and if required by the previous step, the restyling of the interface in order to guarantee its usability.

The evaluation of the usability of the interface is composed by two separate phases: the inspection and the evaluation.

During the inspection phase the interface is analyzed using Nielsen heuristics [2,3,4].

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 422–431, 2010.

The evaluation phase uses the *cognitive walk through* method [5] and the questionnaire to evaluate the usability of the interface.

The results of the two phases are then considered for the production of the final report, that will also benefit from the results of a questionnaire filled by potential final users.

The methodology for the software development is composed by six phases: analysis, design, prototyping, implementation, test and deployment. The various versions are released to the customer as they are produced, according to the *evolutionary deployment* model. Even if the evolutionary deployment is not implied in a evolutionary model, it adds value to the model since it allows to immediately collect feedback from the final user. We define the deployable incremental version a self-contained functional software unit able to perform useful user functions, that is released accompanied by suitable support material (on line documentation, manuals, etc).

2 The USABAGILE_Web Method

The proposed methodology is based on the Agile approach, thus each software interface is defined by a cycle, even if it cannot be used alone, independently from other interfaces. In the following sections we will present the phases on which the method is based.

2.1 Inspection

The first phase of the method is related to the inspection of a particular software component, detecting the usability problems. The analysis is based on Nielsen's heuristics [4].

In the ideal case, the software inspection has to be carried out by a team of 3-5 usability experts.

The detected problems will be ranked (from 1 to 3) according to the following numerical values:

1. Serious problem: it may cause the software to be withdrawn by the user;
2. Intermediate problem: the problem has to be considered and solved
3. Light problem: it can be neglected, if there is lack of time

In the considered case of web-based interfaces, the analysis will be focused on single web page. It is important to point out that in this phase it does not care on the interface's functionality (i. e.: which types of operations the user can perform through it). Only the usability problems related to the interface's structure (problems that may disorient the user or that may cause the abandon of the page) do matter.

2.2 Evaluation

This phase is still managed by the experts, no interactions are necessary with the customers and the end-users.

In this phase the usability of the operations is studied. In our case, in which web pages are considered, the components under analysis are links, forms and all elements the user may interact with.

The adopted methodology for the evaluation phase is the cognitive walk through method that allows to study how the user interacts with the interface.

To perform this study the experts will try to explore the interface functionality having in mind the requirements and needs of a potential user, pointing out which functions are visible and easy to execute and which of them can be removed. The study is necessary to control that the essential interface operations are visible and easy to execute for the user, then to monitor that the execution of such operations receives the feedback from the system and lastly to verify that the most important operations are located in zones of the interface where the user's eyes may be more easily oriented. To this end one has to take into account the outcomes of the studies of the human eye tracking, which studies the privileged movements of the human eyes when a new image appears on a display [6] (in our case when a new interface is opened) indicating that the principal interface functions and information have to be located in the upper area of the interface.

At the end of the analysis the group of experts will decide if the interface has to be rewritten to avoid the detected problems.

For an exhaustive evaluation of the interface usability the group has to know the users' sensations and wishes, after having used it. To collect such information from the users, the method supports a closed questionnaire, which will be described in the following section.

2.3 The Questionnaire

It could happen that some problems in the usability of the interface could not be properly identified by the experts because the physical or the psychological conditions of some members prevent from the identification of the errors. To this end we defined a questionnaire to collect the software's evaluation made by typical users after having used the interface. In this way we provided to the experts a valid support, to integrate the evaluation they have made. Some problems identified by the experts will be emphasized by the analysis of the users' responses to the questionnaire. In other cases the responses will suggest to the experts new approaches to identify problems they do not have taken into account previously. The methodology hitherto described is complete and efficient, since it uses guidelines and approaches well known and consolidated and allows to explore all relevant aspects of the interface, like the graphic design and the operations available to the users. Furthermore the questionnaire allows to perform the task in collaboration with a community of users, according to the agile usability approach.

2.4 Analysis

The agile development methodology starts with the *Analysis* phase. The purpose of such phase is to show the interface behavior applying the Unified Modeling

Language (UML) [7] Use Case Diagrams (UCD) [8], a type of behavioral diagram defined by and created from a Use-case analysis.

In the subsequent phase a prototype (low, medium and high adherence) of the behavior of the new component is shown to the customer. If the interface does not need modifications to its behavior, the customer is informed that no modifications have been made with respect to the status before the analysis phase started.

2.5 Design

The Design phase performs the logical check of operations the user can perform using the interface.

We have to summarize the interface structure defining the actions a user can perform through it. To this purpose we used another UML instrument: the sequence diagrams [7]. The sequence diagrams are able to show the sequence of messages passed between objects in a named run we are considering. These objects can be the actors of the use-case diagrams, classes belonging to the program, links and components (text input, combo box, buttons) of Web interfaces. The sequence diagrams are helpful also to show to the customer how to structure the application and to generate test cases.

The main components of a sequence diagram are the following: *actor*, *object*, *lifeline* and *message*. The actors represents the customer, the administrator or the user interacting with the software. The object is the software component with which one interacts, it can be an instance of a class (Object Oriented programs), a web link; it is represented graphically as a box containing the label representing the instance, according to the following syntax: *instance:Classname*. The lifeline is represented by a dotted line as the time axis. On it the messages between objects are exchanged. These messages can be classified as: *simple*, *synchronous*, *asynchronous*, and *answer*.

A simple message transfers the control of the application from an object to another. A synchronous message blocks the system until a response to the message will be received. An asynchronous message returns immediately the control to the object that have sent the message. An answer message is associated to a response to a synchronous message. The corresponding symbols associated to them are respectively: an arrow with an empty tip, an arrow with a filled tip, an arrow with an empty tip sideways cut, and a dotted arrow.

A self–delegation message is addressed to the object itself: it is mainly used in recursive procedures. A message can contain the stereotype `<create>` if the destination object is created with the message. Similarly, the message can contain the stereotype `<destroy>` if the message will imply the object death.

In a sequence diagram time flows from the top to the bottom, and the transmission time is neglected. It is possible to trace the duration of the operations recording it on the diagram, or labeling messages and adding some notes. Furthermore, it is possible to add some control information, like the *message conditions* (in such case the program continue if a boolean condition is true) or some *iterators* (indicating the multiple dispatch of a message).

The sequence diagrams are not sufficient to figure out the logical behavior of the interface. We have to take into account the possible interfaces we can reach from the component and if are present some data processing techniques or database.

The navigation tree technique is based on a connected graph having as root the interface we are studying and as leaves the interfaces one can reach, so that we can show to the customer the navigation capabilities of the named interface. As per the databases, we take benefit of the object oriented software engineering class diagrams.

At the end of the design phase, the group face with the customer, deciding if the interface design assessment is enough good or not. If not, the design phase is restarted taking into account the outcomes of the assessment.

2.6 Prototyping

The prototyping phase is integrated in the Analysis and Design phases. In fact, as a consequence of the decision made by the group of experts analyzing the behavior of the interface, a new prototype may be implemented on the basis of the input from the group of experts. When a prototype has been implemented it has to be shown to the customer, according to the Agile approach, which demand a constant dialogue with the costumer.

When the various prototypes will be accepted by the customer and by the group of experts, the implementation phase will be activated.

2.7 Implementation

When the prototype related to the interface behavior related to the Design phase, and that implemented in the Design phase to describe the interface functionality, have been accepted by the customer and the group of experts, the new interface can be implemented, producing the related code. This phase is similar to the classic implementation, since do not demand the dialog with the customer, since she/he is not an expert programmer.

The implementation do not requires additional clarifications. It will be a mere phase related to the writing of the code and the timetable for its completion depends only on the skill and the attitude of programmers.

2.8 Test

The test phase is one of the most important phases. The group of developers has to select a set of typical potential users to test the functionality of the new interface. The test has to demonstrate if some improvements have been made during the reverse engineering process. It is preferred to select also some of the users that filled the questionnaire to compare the behavior of the old interface with that of the new one. This is the best practice to be sure that the reengineering process has been successful.

If the test has been unsuccessful the full cycle has to be repeated, having as new inputs the errors or the weakness highlighted in the test phase.

If the test of the interface has been successful and new components has to be implemented, the process continues until the product is complete.

Once the product has been fully implemented and tested, the release process has to be performed.

2.9 Release

The last phase of the method is executed only once, after all development cycles have been successfully completed.

With the release phase the software is definitely validated by the experts and the customer.

3 Full Cycle

The various phases of the USABAGILE_Web method for the software reengineering have to be combined to produce a full cycle as efficient as possible.

In Figure 1 the scheme of the method is sketched. The software to be reengineered is divided in separated components, each of them will be the actor of one or more cycles.

A cycle starts with the Analysis phase, where the component's behavior is analysed using the Use Case Diagrams. At the same time the component starts the Usability inspection phase, to identify the potential problems using Nielsen's Heuristics.

After having completed the Analysis phase, the experts evaluate if it is required to redesign the component behavior, producing in such case a prototype of the new structure, taking into account the usability rules. This mini-prototyping phase is made being in touch with the customer, to figure out her/his requests.

The Design phase is articulated in the same way as the Analysis phase. The logical structure of the operations of the components is studied by means of the Sequence Diagrams, the Navigation Trees and the analysis of the data processing. At the same time the Usability analysis is carried out, studying the user interactions with the component, adopting the cognitive simulation approach [5]. This study will help the experts do determine if a new prototype of the component functionality is required. The dialogue with user is still an important prerequisite for a successful completion of this phase.

The Prototyping phase occurs only if one of the previous two phases required a new prototype. In such case the new version is shown to the customer, taking into account the Nielsen's heuristics concerning the usability aspects.

The Implementation phase is related only to the code implementation and do not require any interaction with the customer.

The Test phase demands an intensive interaction with the customer to test the functionality of the interface, to detect all problems and errors in the interface, including the usability issues.

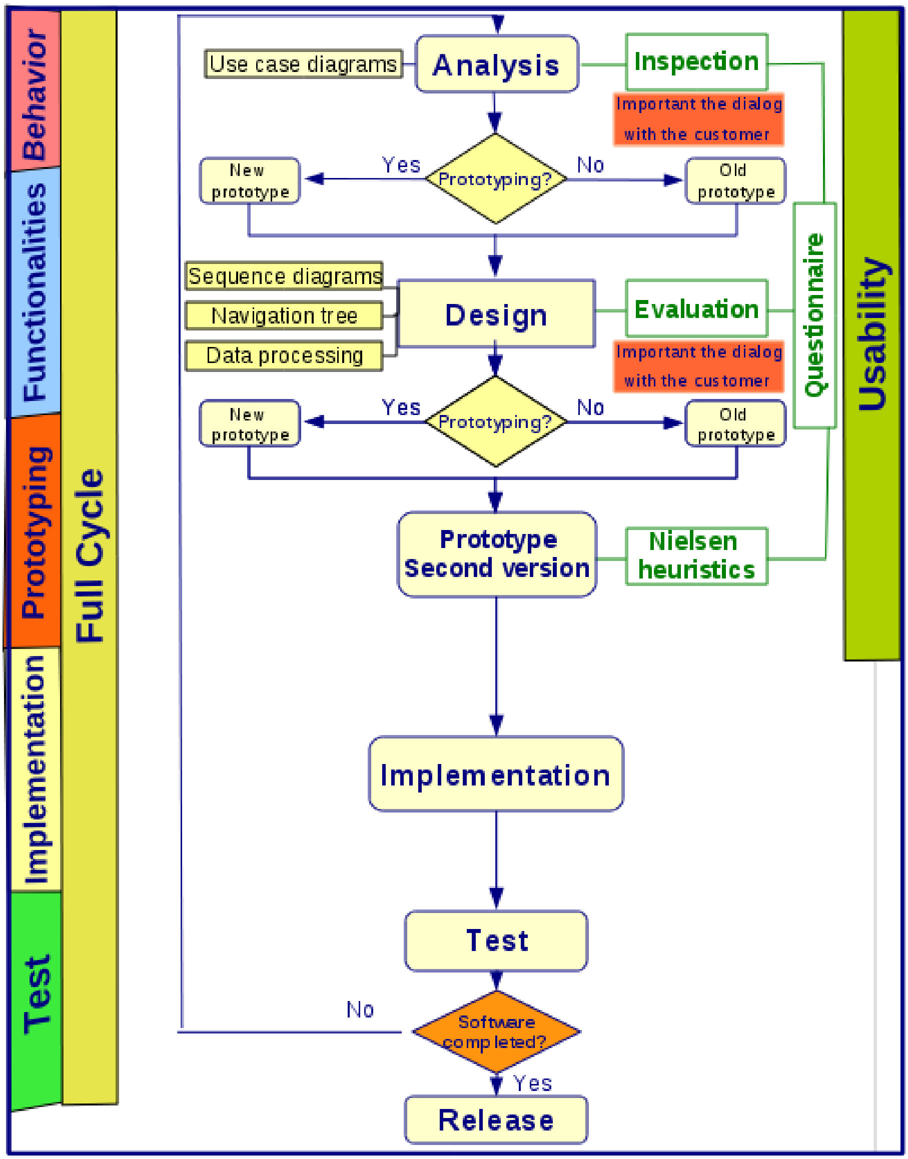


Fig. 1. Scheme of the USABAGILE_Web method

If the Test phase produced satisfactory results, the component is considered completed, otherwise the full cycle restarts taking into account the problems pointed out in the Test phase, until a successful completion of the cycle is obtained.

If the cycle ended successfully and all components have been analysed the cycle is closed and one gets on to the Release phase.

4 Costs

The cost of a software product originates from various sources. Direct costs are related to the following items: the technical staff, the users involved in the test phase, the computational resources, and the general costs. Indirect costs are related to the unavailability of the application, and the backlog.

The forecast of costs are settled by the following elements: the human factor, complexity of the implemented software, stability of requisites (modifications required after the Design phase has been completed), number of iterative cycles used, Test.

The software cost has to be carefully evaluated. This process is articulated in five steps:

a) Formulation: evaluation of suitable measures[1] and metrics[2] for the goal to be reached;
b) Collection: action of collecting data;
c) Analysis: computation of the metrics by using mathematical tools;
d) Interpretation: evaluation of the measurement results
e) Feedback obtained from the measurement results.

Several methods are available to determine the most suitable metrics [9] for the evaluation of software costs, however such cost has to be determined taking into account how are organized the Implementation and Test phases.

5 The Design of Web Interfaces

In a web site we can consider as software components the various XHTML web pages. For each web page we will apply one or more cycles.

The server-side and client side scripts are called from the web pages, so they can be analysed during the Design phase, in the data processing step, instead of consider them as independent components.

In the Design phase the navigation trees have to follow the links present in the various pages of the site e the data processing phase has to study the various scripts present in the various pages. The Usability test is easy to perform on web pages and the cognitive simulation can be easily used to examine them [10].

The remaining phases can be performed easily, according to the described method. The required knowledge is limited to the XHTML and the scripting languages, like PHP and Java. The Test phase can be organized in a way that it will be possible to check both the physical behavior of the web site, using some techniques of test-case generation, and its usability, asking to some potential users to interact with the web site.

[1] A Measure (Software Engineering) is the result of the measurement of a specific attribute.

[2] A software metric is a measure of some property of a piece of software or its specifications (source: Wikipedia).

6 Conclusions

The USABAGILE_Web method is suitable to perform the reverse engineering of any running software and as Agile methodology for designing a new web software application. The method performs in parallel both to the software development and the usability evaluation, activating a dialog with the customer, and testing it involving a group of potential users.

This user centric approach imply very important social effects. In fact, despite the technical insights of the software development, the method innovates the criteria of software evaluation, prioritizing the usability aspects of the product developed and providing to the users some instruments, like the questionnaire, that enable them to effectively express their needs and preferences.

The method also addresses some economical issues. In fact, being centered on the usability of software components, the implemented software products may attract potential users and facilitate the dissemination of the implemented product. The globalization of the economy makes the purchase of hardware and software products more affordable. In this context products developed using the USABAGILE_Web method may spread over the market more easily.

We are confident that also some moral issues are involved in the USABAGILE_Web approach. In fact, facilitating the user involvement, the experts may acquire new capabilities to develop efficient and powerful products, promoting the cultural exchanges, collecting from the final users important requirements. This process will foster the dissemination of innovative technologies and new approaches to make web sites more and more attractive and efficient.

Acknowledgments

Financial support from the the European Union through the COST action D37 (ELAMS working group) is acknowledged.

References

1. Ambler, S.W.: Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process. John Wiley & Sons, New York (2002)
2. Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Francisco (1994)
3. Nielsen, J.: Designing Web Usability: The practice of simplicity. New Riders Publishing, Indianapolis (2000)
4. The description of Nielsen heuristics, http://www.useit.com/papers/heuristic/
5. Nielsen, J., Mack, R.L.: Usability Inspection Methods. John Wiley & Sons, New York (1994)
6. Nielsen, J., Pernice, K.: Eyetracking Web Usability. New Riders Publishing, Berkeley (2010)
7. Booch, G., Rumbaugh, J., Jacobson, I.: Unified Modeling Language User Guide, 2nd edn. Addison-Wesley Object Technology Series. Addison-Wesley Professional, Reading (2005)

8. Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G.: Object oriented software engineering: a use case driven approach. Addison-Wesley, Reading (1992)
9. Capers, J.: Estimating Software costs: Bringing Realism to Estimating, 14th edn., pp. 1462–1477. The Mac Graw Hill Company, New York (2007)
10. Nielsen, J., Tahir, M.: Homepage Usability: 50 Websites Deconstructed. New Riders Publishing, Indianapolis (2001)

Combining CSP and Constraint-Based Mining for Pattern Discovery

Mehdi Khiari, Patrice Boizumault, and Bruno Crémilleux

GREYC, Université de Caen Basse-Normandie, Campus Côte de Nacre,
F-14032 Caen Cedex, France
{Forename.Surname}@info.unicaen.fr

Abstract. A well-known limitation of a lot of data mining methods is the huge number of patterns which are discovered: these large outputs hamper the individual and global analysis performed by the end-users of data. That is why discovering patterns of higher level is an active research field. In this paper, we investigate the relationship between local constraint-based mining and constraint satisfaction problems and we propose an approach to model and mine patterns combining several local patterns, i.e., patterns defined by n-ary constraints. The user specifies a set of n-ary constraints and a constraint solver generates the whole set of solutions. Our approach takes benefit from the recent progress on mining local patterns by pushing with a solver on local patterns all local constraints which can be inferred from the n-ary ones. This approach enables us to model in a flexible way *any* set of constraints combining several local patterns. Experiments show the feasibility of our approach.

1 Introduction

In current scientific, industrial or business areas, the critical need is not to generate data, but to derive knowledge from huge datasets produced at high throughput. Extracting or discovering knowledge from large amounts of data is at the core of the Knowledge Discovery in Databases task, often also named "data mining". This involves different challenges, such as designing efficient tools to tackle data and the discovery of patterns of a potential user's interest. There is a large range of methods to discover the patterns but it is well-known that the "pattern flooding which follows data flooding" is an unfortunate consequence in exploratory Knowledge Discovery in Databases processes and the most significant patterns are lost among too much trivial, noisy and redundant information.

Many works propose methods to reduce the collection of patterns, such as the constraint-based paradigm [23], the pattern set discovery approach [9,17], the so-called condensed representations [5] as well as the compression of the dataset by exploiting the Minimum Description Length Principle [25]. The constraint-based pattern mining framework is a powerful paradigm to discover new highly valuable knowledge [23]. Constraints provide a focus on the most promising knowledge by reducing the number of extracted patterns to those of potential interest for user. There are now generic approaches to discover *local patterns*

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 432–447, 2010.

(cf. Section 2.1) under constraints [8,26] and this issue is rather well-mastered, at least for data described by items (i.e., boolean attributes). We call *local constraints* the constraints addressing local patterns. Here, locality refers to the fact that checking whether a pattern satisfies or not a constraint can be performed independently of the other patterns holding in the data. Nevertheless, even if the number of produced local patterns is reduced thanks to the constraint, the output still remains too large for individual and global analysis by the end-user.

On the other hand, the interest of a pattern also depends on the other patterns which are mined. A lot of patterns which are expected by the user (cf. Section 2.2) or models such as classifiers or clustering require to consider simultaneously several patterns to combine the fragmented information conveyed by the local patterns. Local constraints, by considering only one pattern, are insufficient to define and discover such higher patterns. There are few attempts on particular cases by using devoted methods [28,18] but there is no generic approach. That is why we claim that discovering patterns under constraints involving comparisons between local patterns is a major issue. In the following of this paper, we call *n-ary constraints* such constraints.

Mining patterns under local constraints requires the exploration of a large search space, even in the case of the simplest patterns, i.e., data described by items. Obviously, mining patterns under n-ary constraints is even harder because we have to take into account and compare the solutions satisfying each pattern involved in a n-ary constraint. In this paper, we investigate the relationship between constraint-based mining and constraint programming and we propose an approach to model and mine patterns under n-ary constraints. As Constraint Satisfaction Problem (CSP) has the ability to define constraints on several variables [1], it is a natural way to model n-ary constraints. We show that each pattern of a n-ary constraint can be assimilated to a variable in the CSP framework. The great advantage of this modeling is its flexibility, it enables us to define a large broad of n-ary constraints. Basically, with our approach, the user specifies the model, that is, the set of n-ary constraints which has to be satisfied, and a constraint solver generates the correct and complete set of solutions. The CSP community has developed several efficient constraint solvers that we can reuse and the resolution can be performed at the level of this global modeling. But we think that it would be a pity not to take benefit from the recent progress on mining local patterns. That is why a key point of our approach is to divide a n-ary constraint in two parts, i.e., a set of local constraints $\mathcal{C}_{loc}$ which is solved by a solver on local patterns and a set of n-ary constraints $\mathcal{C}_{n-ary}$ which is solved by a CSP solver (cf. Section 4 for more details). We claim that is this combination between the local and n-ary levels which enables us the discovery of patterns under n-ary constraints. In other words, the contribution of this paper is to propose an approach joining local constraint mining and set constraint programming in order to model n-ary constraints and discover patterns under such constraints. More generally, the paper investigates the relationship between constraint-based mining and set constraint programming.

This paper is organized as follows. Section 2 sketches definitions and presents the problem statement. The background on pattern discovery and set constraint programming is given in Section 3. We propose our approach to model and mine patterns under n-ary constraints in Section 4. Section 5 details experiments and deals with a discussion and research issues related to our approach.

2 Definitions and Motivations

Below we give definitions used in the paper and the context and motivations.

2.1 Definitions

Let $\mathcal{I}$ be a set of distinct literals called *items*, an itemset (or pattern) is a non-null subset of $\mathcal{I}$. The language of itemsets corresponds to $\mathcal{L}_{\mathcal{I}} = 2^{\mathcal{I}} \backslash \emptyset$. A transactional dataset is a multi-set of itemsets of $\mathcal{L}_{\mathcal{I}}$. Each itemset, usually called transaction or object, is a database entry. For instance, Table 1 gives a transactional dataset $\boldsymbol{r}$ where 9 objects $o_1, \ldots, o_9$ are described by 6 items $A, \ldots, c_2$.

Table 1. Example of a transactional context $\boldsymbol{r}$

Trans.	Items					
o_1	A	B			c_1	
o_2	A	B			c_1	
o_3			C		c_1	
o_4			C		c_1	
o_5			C		c_1	
o_6	A	B	C	D		c_2
o_7			C	D		c_2
o_8			C			c_2
o_9				D		c_2

Let X be a local pattern. Pattern mining aims at discovering information from all the patterns or a subset of $\mathcal{L}_{\mathcal{I}}$. More precise, constraint-based mining task selects all the itemsets of $\mathcal{L}_{\mathcal{I}}$ present in $\boldsymbol{r}$ and satisfying a predicate which is named *constraint*. Local patterns are regularities that hold for a particular part of the data. A local pattern is of special interest if it exhibits a deviating behavior w.r.t. the underlying global model of the data [14] because we are seeking for surprising knowledge which deviates from the already known background model. There are a lot of constraints to evaluate the relevance of local patterns. A well-known example is the frequency constraint which focuses on patterns having a frequency in the database exceeding a given minimal threshold $\gamma > 0$: $\mathit{freq}(X) \geq \gamma$. Many works [23] replace the frequency by other interestingness measures to evaluate the relevance of patterns such as the area of a pattern ($\mathit{area}(X)$ is the product of the frequency of the pattern times its length, i.e., $\mathit{area}(X) = \mathit{freq}(X) \times \mathit{count}(X)$ where $\mathit{count}(X)$ denotes the cardinality of X).

In practice, the user is often interested in discovering more complex patterns such as the simplest rules in the classification task based on associations [30], pairs of exception rules [28] which may reveal global characteristics from the database. The definition of such patterns relies on properties involving several local patterns [6]. These patterns are formalized by the notion of *n-ary constraint*:

Definition 1 (n-ary constraint). *A constraint q is said n-ary if several local patterns have to be compared to check if q is satisfied or not.*

The next section provides more precise examples of n-ary constraints.

2.2 Context and Motivations

N-ary constraints are very useful to design a lot of patterns requested by the users. For instance, the discovery of exception rules from a data set without domain-specific information is of a great interest [28]. An exception rule is defined as a deviational pattern to a strong rule and the interest of an exception rule is evaluated according to another rule. The comparison between rules means that these exception rules are not local patterns. More formally, an exception rule is defined within the context of a pair of rules as follows (I is an item, for instance a class value, X and Y are local patterns):

$$exception(X \rightarrow \neg I) \equiv \begin{cases} true \quad \text{if } \exists Y \in \mathcal{L}_{\mathcal{I}} \text{ such that } Y \subset X \text{, one have} \\ \qquad\qquad\qquad\qquad\qquad\qquad (X \backslash Y \rightarrow I) \wedge (X \rightarrow \neg I) \\ false \text{ otherwise} \end{cases}$$

Such a pair of rules is composed of a common sense rule $X \backslash Y \rightarrow I$ (the term “common sense rule” represents a user-given belief) and an exception rule $X \rightarrow \neg I$ since usually if $X \backslash Y$ then I. The exception rule isolates unexpected information. This definition assumes that the common sense rule has a high frequency and a rather high confidence and the exception rule has a low frequency and a very high confidence (the confidence of a rule $X \rightarrow Y$ is $\mathit{freq}(X \cup Y)/\mathit{freq}(X)$). Assuming that a rule $X \rightarrow Y$ holds iff at least 2/3 of the transactions containing X also contains Y, the rule $AC \rightarrow \neg c_1$ is an exception rule in our running example (cf. Table 1) because we jointly have $A \rightarrow c_1$ and $AC \rightarrow \neg c_1$. Note that Suzuki proposes a method based on sound pruning and probabilistic estimation [28] to extract the exception rules. Nevertheless, this method is devoted to this kind of patterns.

In the context of genomics, local patterns defined by groups of genes and satisfying the *area* constraint previously introduced above are at the core of the discovery of synexpression groups [15]. Nevertheless, in noisy data such as transcriptomic data, the search of fault-tolerant patterns is very useful to cope with the intrinsic uncertainty embedded in the data [3]. N-ary constraints are a way to design such fault-tolerant patterns: larger sets of genes with few exceptions are expressed by the union of several local patterns satisfying an area constraint and having a large overlapping between them. From two local patterns, it corresponds to the following n-ary constraint: $area(X) > min_{area} \wedge area(Y) > min_{area} \wedge (area(X \cap Y) > \alpha \times min_{area})$ where min_{area} denotes the minimal area and α is a threshold given by the user to fix the minimal overlapping between the local patterns. The set of n-ary constraints can also be extended by the use of the universal quantifier (see Section 6).

Section 4 presents our approach to model patterns satisfying such n-ary constraints and how we combine local constraint mining and set constraint programming to extract these patterns.

3 Background: Related Works and Set CSP

3.1 Local Patterns and Pattern Sets Discovery

As said in the introduction, there are a lot of works to discover local patterns under constraints. A key issue of these works is the use of the property of monotonicity because pruning conditions are straightforwardly deduced [21]. A constraint q is anti-monotone w.r.t. the item specialization iff for all $X \in \mathcal{L}_{\mathcal{I}}$ satisfying q, any subset of X also satisfies q. In this paper, we use the Music-dfs[1] prototype because it offers a set of syntactic and aggregate primitives to specify a broad spectrum of constraints in a flexible way [27]. Music-dfs mines soundly and completely all the patterns satisfying a given set of input local constraints. The efficiency of Music-dfs lies in its depth-first search strategy and a safe pruning of the pattern space exploiting the anti-monotonicity property to push the local constraints as early as possible. The pruning conditions are based on intervals representing several local patterns. The local patterns satisfying all the local constraints are provided in a condensed representation made of intervals (each interval represents a set of patterns satisfying the constraint and each pattern appears in only one interval). The lower bound of an interval is a prefix-free pattern and its upper bound is the prefix-closure of the lower bound [27].

There are also other approaches to combine local patterns. Recent approaches - pattern teams [17], constraint-based pattern set mining [9] and selecting patterns according to the added value of a new pattern given the currently selected patterns [4] - aim at reducing the redundancy by selecting patterns from the initial large set of local patterns on the basis of their usefulness in the context of the other selected patterns. Even if these approaches explicitly compare patterns between them, they are mainly based on the reduction of the redundancy or specific aims such as classification processes. We think that n-ary constraints are a flexible way to take into account a bias given by the user to direct the final set of patterns toward a specific aim such as the search of exceptions. General data mining frameworks based on the notion of local patterns to design global models are presented in [16,13]. These frameworks help to analyze and improve current methods in the area. In our approach (cf. Section 4), we show the interest of the set constraint programming in this general issue of combining local patterns.

[1] http://www.info.univ-tours.fr/~soulet/music-dfs/music-dfs.html

Constraint programming is a powerful declarative paradigm for solving difficult combinatorial problems. In a constraint programming approach, one specifies constraints on acceptable solutions and search is used to find a solution that satisfies the constraints. A first approach using Constraint Programming for itemset mining has been proposed in [7]. In this work, constraints such as frequency, closedness, maximality, and constraints that are monotonic or anti-monotonic or variations of these constraints are modeled using 0/1 Linear Programming. Then patterns satisfying these constraints are obtained by using the constraint solver Gecode [11]. This work presents in a unified framework a large set of patterns but does not address patterns modeled by relationships between several local patterns as those described in Section 2. Recently, this work has been extended in order to find correlated patterns (i.e., patterns having the highest score w.r.t. a correlation measure) [24].

3.2 Set CSP

Formally a Constraint Satisfaction Problem (CSP) is a 3-uple $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ where $\mathcal{X}$ is a set of variables, $\mathcal{D}$ is a set of finite domains and $\mathcal{C}$ is a set of constraints that restrict certain simultaneous variables assignments. There are several types of CSPs such as numerical CSPs, boolean CSPs, set CSPs, etc. They differ fundamentally from the domain types and filtering techniques. We present here more precisely set CSPs that are used in our modeling. First, we define Set Intervals. Then we introduce set CSPs, and give an example. Finally we present some filtering rules for set CSPs.

Definition 2 (Set Interval). *let lb and ub be two sets such that $lb \subset ub$, the set interval $[lb..ub]$ is defined as follows: $[lb..ub] = \{E$ such that $lb \subseteq E$ and $E \subseteq ub\}$.*

Set intervals avoid data storage problems due to the size of domains: they model the domain and encapsulate all the possible values of the variables. For example: $[\{1\}..\{1,2,3\}]$ summarizes $\{\{1\},\{1,2\},\{1,3\},\{1,2,3\}\}$ and $[\{\}..\{1,2,3\}]$ summarizes $2^{\{1,2,3\}}$.

Definition 3 (Set CSP). *A set constraint satisfaction problem (set CSP) is a 3-uple $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ where $\mathcal{C} = \{c_1, ..., c_m\}$ is a set of constraints associated to a set $\mathcal{X} = \{X_1, ..., X_n\}$ of variables. For each variable X_i, an initial domain of set intervals (or union of set intervals) D_{X_i} is given and $D = \{D_{X_i}, ..., D_{X_n}\}$.*

In order to illustrate the declarative feature and the expressiveness of set CSPs, we give the following example.

Example. [29] Two transmitters have to be assigned to two radio frequencies each. Available frequencies are $\{1,2,3,4\}$ for the first transmitter and $\{3,4,5,6\}$ for the second one. The distance between these two frequencies is equal to the absolute value of the difference between these frequencies. The constraints are:

- two radio frequencies have to be assigned to each transmitter: $c_1 \wedge c_2$.
- both transmitters do not share frequencies: c_3
- two frequencies within a transmitter must have at least a distance equals to 2: c_4
- the first transmitter requires the frequency 3: c_5
- the second transmitter requires the frequency 4: c_6

It can be expressed as a set CSP $(\mathcal{X}, \mathcal{D}, \mathcal{C})$, where:

- $\mathcal{X} = \{t_1, t_2\}$ where t_1 and t_2 are the two transmitters.
- $D(t_1) = [\{\} \,..\, \{1,2,3,4\}]$ and $D(t_2) = [\{\} \,..\, \{3,4,5,6\}]$.
- $\mathcal{C} = \{c_1, c_2, c_3, c_4, c_5, c_6\}$ where:
  - $c_1 \quad \mid t_1 \mid = 2$
  - $c_2 \quad \mid t_2 \mid = 2$
  - $c_3 \quad t_1 \cap t_2 = \emptyset$
  - $c_4 \quad \forall v_1, v_2 \in t_i, \quad \mid v_1 - v_2 \mid \geq 2 \quad i = 1,2$
  - $c_5 \quad 3 \in t_1$
  - $c_6 \quad 4 \in t_2$

This problem has a unique solution where the first transmitter is assigned to the frequencies $\{1,3\}$ and the second to $\{4,6\}$.

Examples of Filtering Rules for Set CSPs. For CSPs, filtering consists on reducing the variable domains in order to remove values that cannot occur in any solution. As soon as a domain D_{X_i} becomes empty (i.e., there is no available value for X_i), a failure is generated for the search. Filtering rules for integer intervals and set intervals are presented in [22,19,12]. We now present two examples of filtering rules for set intervals, the *inclusion* and the *intersection* constraints:

Let $D_x = [a_x .. b_x]$, $D_y = [a_y \,..\, b_y]$ and $D_z = [a_z \,..\, b_z]$ three domains represented by set intervals and D'_x, D'_y and D'_z the filtered domains.

- **Constraint:** $X \subset Y$

  Filtering rule:
  ```
  if $a_x \subset b_y$ then
      $D'_x = [a_x \,..\, b_x \cap b_y]$
      $D'_y = [a_x \cup a_y \,..\, b_y]$
  else
      $D'_x = \emptyset, D'_y = \emptyset$
  ```

- **Constraint:** $Z = X \cap Y$

  Filtering rule:
  ```
  if $(b_x \cap b_y) \subset b_z$ and $(b_x \cap b_y) \neq \emptyset$ then
      $D'_x = [a_x \cup a_z \,..\, b_x \setminus ((b_x \cap a_y) \setminus b_z]$
      $D'_y = [a_y \cup a_z \,..\, b_y \setminus ((b_y \cap a_x) \setminus b_z]$
      $D'_z = [a_z \cup (a_x \cap a_y) \,..\, b_z \cap b_x \cap b_y]$
  else
      $D'_x = D'_y = D'_z = \emptyset$
  ```

Programming Tool: ECL^iPS^e. [10] is a Constraint Programming Tool supporting the most common techniques used in solving constraints satisfaction (or optimization) problems: Constraint Satisfaction Problems, Mathematical Programming, Local Search and combinations of those. ECL^iPS^e is built around the Constraint Logic Programming paradigm [1]. Different domains of constraints as numeric CSP and Set CSPs can be used together. Finally, libraries for solving set CSPs, as *ic-sets* or *conjunto* [12], are available in ECL^iPS^e.

4 Set Constraint Programming for Pattern Discovery

Our approach is based on two major points. First, we use the wide possibilities of modelization and resolution given by the CSPs, in particular the set CSPs and numeric CSPs. Second, we take benefit from the recent progress on mining local patterns. The last choice is also strengthened by the fact that local constraints can be solved before and regardless n-ary constraints.

In this section, we start by giving an overview of our approach. Then we describe each of the three steps of our method by considering the example of the exception rules described in Section 2.2.

4.1 General Overview

Figure 1 provides a general overview of the three steps of our approach:

1. Modeling the query as CSPs, then splitting constraints into local ones and n-ary ones.
2. Solving local constraints using a local pattern extractor (Music-dfs, introduced in Section 3.1) which produces an interval condensed representation of all patterns satisfying the local constraints.
3. Solving n-ary constraints of the CSPs by using ECL^iPS^e (introduced in Section 3.2) where the domain of each variable results from the interval condensed representation (computed in the Step-2).

4.2 Step-1: Modelling the Query as CSPs

Let $\boldsymbol{r}$ be a dataset having nb transactions, and $\mathcal{I}$ the set of all its items. We model the problem by using two CSPs $\mathcal{P}$ and $\mathcal{P}'$ that are inter-related:

1. Set CSP $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ where:
   - $\mathcal{X} = \{X_1, ..., X_n\}$. Each variable X_i represents an unknown itemset.
   - $\mathcal{D} = \{D_{X_1}, ..., D_{X_n}\}$. The initial domain of each variable X_i is the set interval $[\{\} \mathinner{..} \mathcal{I}]$.
   - $\mathcal{C}$ is a conjunction of set constraints by using set operators ($\cup$, $\cap$, $\setminus$, $\in$, $\notin$, ...)
2. Numeric CSP $\mathcal{P}' = (\mathcal{F}, \mathcal{D}', \mathcal{C}')$ where:
   - $\mathcal{F} = \{F_1, ..., F_n\}$. Each variable F_i is the frequency of the itemset X_i.

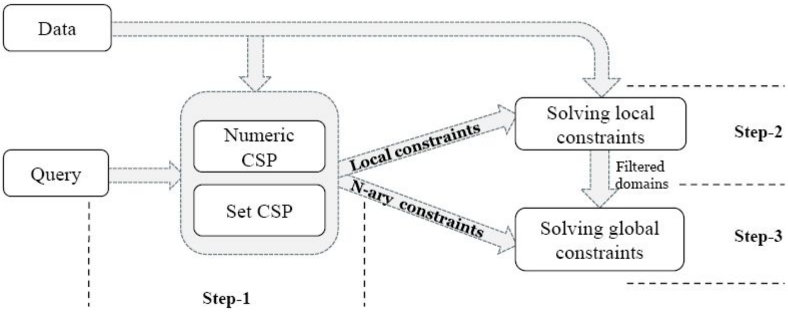


Fig. 1. General overview of our 3-steps method

- $\mathcal{D}' = \{D_{F_1}, ..., D_{F_n}\}$. The initial domain of each variable F_i is the integer interval $[1 .. nb]$.
- $\mathcal{C}'$ is a conjunction of arithmetic constraints.

Then, the whole set of constraints $(\mathcal{C} \cup \mathcal{C}')$ is divided into two subsets as follows:

- $\mathcal{C}_{loc}$ is the set of local constraints to be solved (by Music-dfs). Solutions are given in the form of an interval condensed representation.
- $\mathcal{C}_{n-ary}$ is the set of n-ary constraints to be solved (by ECL^iPS^e), where the domain of the variables X_i and F_i will be deduced from the interval condensed representation computed in the previous step.

Local (unary) constraints can be solved before and regardless n-ary constraints. The search space of the n-ary constraints is reduced by the space of solutions satisfying local constraints. This ensures that every solution verifies both local and n-ary constraints.

4.3 Example: Modeling the Exception Rules as CSPs

Recall that the definition of the pairs of exception rules is given in Section 2.2.

Reformulation: Let $freq(X)$ be the frequency value of the itemset X. Let I and $\neg I \in \mathcal{I}$ (in this example, I and $\neg I$ represent the two class values of the data set). Let $\gamma_1, \gamma_2, \delta_1, \delta_2 \in \mathbb{N}$. The exception rules constraint can be formulated as it follows:

- $X \backslash Y \rightarrow I$ can be expressed by the conjunction: $freq((X \setminus Y) \sqcup$[2]$I) \geq \gamma_1 \ \wedge$ $(freq(X \setminus Y) - freq((X \setminus Y) \sqcup I)) \leq \delta_1$ which means that $X \backslash Y \rightarrow I$ must be a frequent rule having a high confidence value.

[2] The symbol $\sqcup$ denotes the disjoint union operator.

- $X \rightarrow \neg I$ can be expressed by the conjunction: $\mathit{freq}(X \sqcup \neg I) \leq \gamma_2 \wedge (\mathit{freq}(X) - \mathit{freq}(X \sqcup \neg I)) \leq \delta_2$ which means that $X \rightarrow \neg I$ must be a rare rule having a high confidence value.

To sum up:

$$\mathit{exception}(X \rightarrow \neg I) \equiv \begin{cases} \exists Y \subset X \text{ such that:} \\ \mathit{freq}((X \setminus Y) \sqcup I) \geq \gamma_1 \wedge \\ (\mathit{freq}(X \setminus Y) - \mathit{freq}((X \setminus Y) \sqcup I)) \leq \delta_1 \wedge \\ \mathit{freq}(X \sqcup \neg I) \leq \gamma_2 \wedge \\ (\mathit{freq}(X) - \mathit{freq}(X \sqcup \neg I)) \leq \delta_2 \end{cases}$$

CSP Modelisation: The CSP variables are defined as follows:

- Set variables $\{X_1, X_2, X_3, X_4\}$ representing unknown itemsets:
  - $X_1 : X \setminus Y$,
  - $X_2 : (X \setminus Y) \sqcup I$ (common sense rule),
  - $X_3 : X$,
  - $X_4 : X \sqcup \neg I$ (exception rule).
- Integer variables $\{F_1, F_2, F_3, F_4\}$ representing their frequency values (variable F_i denotes the frequency of the itemset X_i).

Table 2 provides the constraints modeling the exception rules.

Table 2. Exception rules modeled as CSP constraints

Constraints	CSP formulation	Local	N-ary
$\mathit{freq}((X \setminus Y) \sqcup I) \geq \gamma_1$	$F_2 \geq \gamma_1$	×	
	$\wedge \quad I \in X_2$	×	
	$\wedge \quad X_1 \subsetneq X_3$		×
$\mathit{freq}(X \setminus Y) - \mathit{freq}((X \setminus Y) \sqcup I) \leq \delta_1$	$F_1 - F_2 \leq \delta_1$		×
	$\wedge \quad X_2 = X_1 \sqcup I$		×
$\mathit{freq}(X \sqcup \neg I) \leq \gamma_2$	$F_4 \leq \gamma_2$	×	
	$\wedge \quad \neg I \in X_4$	×	
$\mathit{freq}(X) - \mathit{freq}(X \sqcup \neg I) \leq \delta_2$	$F_3 - F_4 \leq \delta_2$		×
	$\wedge \quad X_4 = X_3 \sqcup \neg I$		×

Summary:

- Set CSP
  - $\mathcal{X} = \{X_1, ..., X_4\}$
  - $\mathcal{C} = \{(I \in X_2), (X_2 = X_1 \sqcup I), (\neg I \in X_4), (X_4 = X_3 \sqcup \neg I), (X_1 \subsetneq X_3\})$
- Numeric CSP
  - $\mathcal{F} = \{F_1, ..., F_4\}$
  - $\mathcal{C}' = \{(F_2 \geq \gamma_1), (F_1 - F_2 \leq \delta_1), (F_4 \leq \gamma_2), (F_3 - F_4 \leq \delta_2)\}$
- $\mathcal{C}_{loc} = \{(I \in X_2), (F_2 \geq \gamma_1), (F_4 \leq \gamma_2), (\neg I \in X_4)\}$
- $\mathcal{C}_{n-ary} = \{(F_1 - F_2 \leq \delta_1), (X_2 = X_1 \sqcup I), (F_3 - F_4 \leq \delta_2), (X_4 = X_3 \sqcup \neg I), (X_1 \subsetneq X_3)\}$

4.4 Step-2: Solving Local Constraints

As already said, we use for this task MUSIC-DFS (see Section 3.1) which mines soundly and completely local patterns. In order to fully benefit from the efficiency of the local pattern mining, the set of local constraints $\mathcal{C}_{loc}$ is split into a disjoint union of $\mathcal{C}_i$ (for $i \in [1..n]$) where each $\mathcal{C}_i$ is the set of local constraints related to X_i and F_i. Each C_i can be separately solved. Let CR_i be the interval condensed representation of all the solutions of $\mathcal{C}_i$. $CR_i = \bigcup_p (f_p, I_p)$ where I_p is a set interval verifying: $\forall x \in I_p,\ \mathit{freq}(x) = f_p$. Then the filtered domains (see Section 4.3) for variable X_i and variable F_i are:

- D_{F_i} : the set of all f_p in CR_i
- D_{X_i} : $\bigcup_{I_p \in CR_i} I_p$

Example. Let us consider the dataset $\boldsymbol{r}$ (see Table 1) and the local constraints for the exception rules $\mathcal{C}_{loc} = \{(I \in X_2), (F_2 \geq \gamma_1), (F_4 \leq \gamma_2), (\neg I \in X_4)\}$ (see Section 4.3). The respective values for $(I, \neg I, \gamma_1, \delta_1, \gamma_2, \delta_2)$ are $(c_1, c_2, 2, 1, 1, 0)$. The local constraints set related to X_2 is $\mathcal{C}_2 = \{c_1 \in X_2, F_2 \geq 2\}$ is solved by MUSIC-DFS with the following query showing that the parameters given to MUSIC-DFS are straightforwardly deduced from $\mathcal{C}_{loc}$.

```
---------------
./music-dfs -i donn.bin -q "{c1} subset X2 and freq(X2)>=2;"
X2 in [A, c1]..[A, c1, B ] U [B, c1] -- F2 = 2 ;
X2 in [C, c1] -- F2 = 3
---------------
```

4.5 Step-3: Solving n-ary Constraints

Then, from the condensed representation of all patterns satisfying local constraints, domains of the variables X_i and F_i (for $i \in \{1, 2, 3, 4\}$) are updated.

Given the parameters $I = c_1, \neg I = c_2, \delta_1 = 1$ and $\delta_2 = 0$ ($\gamma_1 = 2$ and $\gamma_2 = 1$ are already used in Step-2) and the data set in Table 1, the following ECL^iPS^e session illustrates how all pairs of exception rules can be obtained by using backtracking:

```
---------------
[eclipse 1]:
?- exceptions(X1, X2, X3, X4).
Sol1 : X1 = [A,B], X2=[A,B,c1], X3=[A,B,C], X4=[A,B,C,c2];
Sol2 : X1 = [A,B], X2=[A,B,c1], X3=[A,B,D], X4=[A,B,D,c2];
.../...
---------------
```

5 Experiments

This section shows the practical usage and the feasibility of our approach. This experimental study is conducted on the *postoperative-patient-data* coming from the UCI machine learning repository[3]. This data set gathers 90 objects described by 23 items and characterized by two classes (two objects of a third class value were put aside). We test our approach by using the exception rules as a n-ary constraint (in the following, we use a class value for the item I given in the definition of an exception rule). As previously said, we use Music-dfs (see Section 3.1) and ECL^iPS^e (see Section 3.2). All the tests were performed on a 2 GHz Intel Centrino Duo processor with Linux operating system and 2GB of RAM memory.

These experiments show the feasibility of our approach. Given $(I, \gamma_1, \delta_1, \gamma_2, \delta_2)$ a set of values, our method is able to mine the correct and complete set of all pairs of exception rules.

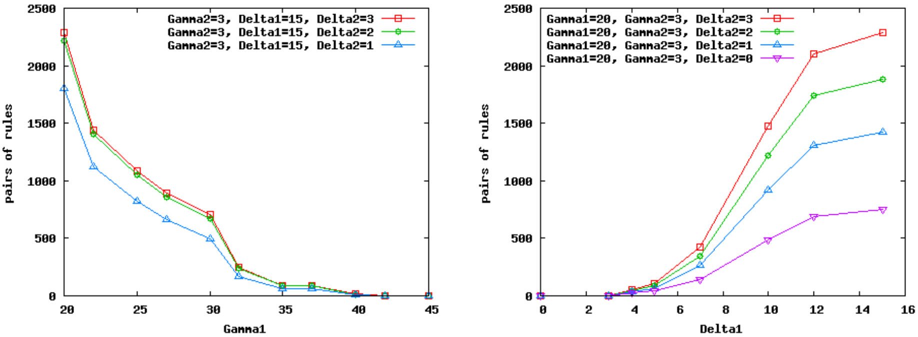


Fig. 2. Number of rules according to γ_1 (left) and δ_1 (right)

Figure 2 depicts the number of pairs of rules according to γ_1 (left part of the figure) and δ_1 (right part of the figure). We tested several combinations of the parameters. As expected, the lower γ_1 is, the larger the number of pairs of exception rules. Note that the decreasing of the curves is approximatively the same for all the combinations of parameters. The result is similar when δ_1 varies (right part of Figure 2): the higher δ_1 is, the larger the number of pairs of exception rules (when δ_1 increases, the confidence decreases so that there are more common sense rules). Interestingly, these curves quantify the number of pairs of exception rules according to the sets of parameters. Some cases seem to point out pairs of rules of good quality. For instance, with $(\gamma_1 = 20, \delta_1 = 5, \gamma_2 = 1, \delta_2 = 0)$, we obtain 25 pairs of rules with a common sense rule having a confidence value greater than or equal to 83% and an exact exception rule (i.e., confidence value equals 100%). Moreover, our approach enables us in a natural way to add new properties such as the control of the sizes of rules. If the user wants that the number of items added to an exception rule remains small with regards to the size of the common sense rule, it can be easily modeled by a new

[3] www.ics.uci.edu/~mlearn/MLRepository.html

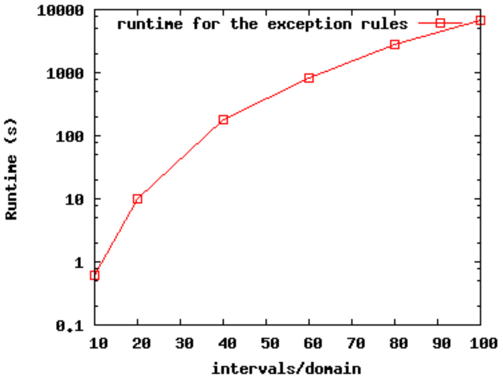


Fig. 3. Runtime according to the number of intervals of the condensed representations

constraint: for instance, the number of added items to an exception rule must be lower than the minimum of a number (e.g., 3) and the size of the common sense rule. It highlights the flexibility of our approach.

Figure 3 details the runtime of our method according to the number of intervals of the condensed representation, i.e., the size of the condensed representation. In this experiment, for each dot of the curve, the four variables have the same domain and thus the same number of intervals. Obviously, the larger the number of intervals is, the higher the runtime (note that we use a logarithmic scale on the Y axis). In the case of exception rules, it is interesting to note that the runtime decreases when the quality of the exception rule pairs increases. Indeed, looking for common sense rules with high frequency and reliable exception rules leads to infer local constraints giving more powerful pruning conditions and thus less intervals.

Table 3 indicates the number of intervals of the variable X_2 in the condensed representation (see Section 4.3) according to several local constraints. It shows the interest of an approach based on local constraint mining.

Discussion. We briefly discuss the set union operator for set CSPs which is a key point in our approach. In order to perform bound consistency filtering, set CSP solvers approximate the union of two set intervals by their convex closure. The convex closure of $[lb_1 \mathinner{..} ub_1]$ and $[lb_2 \mathinner{..} ub_2]$ is defined as $[lb_1 \cap lb_2 \mathinner{..} ub_1 \cup ub_2]$. So, if filtering is applied a lot of times on a same variable domain, this domain may reach the whole set $[\emptyset \mathinner{..} \mathcal{I}]$ and specific information gathered during the search would be lost whereas this information is useful to limit the size of intervals. To circumvent this problem, for each variable X_i with the condensed representation $CR_i = \bigcup_p (f_p, I_p)$, a search is successively performed upon each I_p. This approach is sound and complete and we use it in our experiments. Nevertheless, with this method, we do not fully profit from filtering because removing a value is propagated only in the treated intervals and not in the whole domains. It explains the results of Section 5 showing that the runtime strongly increases when the number

Table 3. Number of intervals according to several local constraints (case of D_{X_2})

Local constraint	Number of intervals in D_{X_2}
-	3002
$I \in X_2$	1029
$I \in X_2 \wedge freq(X_2) >= 20$	52
$I \in X_2 \wedge freq(X_2) >= 25$	32

of intervals increases. Alternative solutions consist of implementing a set interval union operator in the kernel of the solver or using non-exact condensed representations to reduce the number of produced intervals (e.g., a condensed representation based on maximal frequent itemsets). In this case, the number of intervals representing the domains will be smaller, but, due to the approximations, it should be necessary to memorize forbidden values.

6 Conclusion and Future Work

In this paper we have presented a new approach for pattern discovery. Its great interest is to model in a flexible way any set of constraints combining several local patterns. The complete and sound set of patterns satisfying the constraints is mined thanks to a joint cooperation between a solver on set constraint programming which copes with n-ary constraints and a solver on local patterns to take benefit on the well-mastered methods on local constraint mining. We think that it is this combination between the local and n-ary levels which enables us the discovery of such patterns. Experiments show the feasibility of our approach.

In classic CSPs, all variables are existentially quantified. Further work is to introduce the universal quantification (∀): this quantifier would be precious to model important constraints such as the peak constraint (the peak constraint compares neighbor patterns and a peak pattern is a pattern whose all neighbors have a value for a measure lower than a threshold). For that purpose, we think that recent works as Quantified Constraints Satisfaction Problems (QCSP) [2,20] could be useful.

Acknowledgments. The authors would like to thank Arnaud Soulet for very fruitful discussions and Music-dfs prototype. This work is partly supported by the ANR (French Research National Agency) funded project Bingo2 ANR-07-MDCO-014.

References

1. Apt, K.R., Wallace, M.: Constraint Logic Programming using Eclipse. Cambridge University Press, New York (2007)
2. Benhamou, F., Goualard, F.: Universally quantified interval constraints. In: Dechter, R. (ed.) CP 2000. LNCS, vol. 1894, pp. 67–82. Springer, Heidelberg (2000)

3. Besson, J., Robardet, C., Boulicaut, J.-F.: Mining a new fault-tolerant pattern type as an alternative to formal concept discovery. In: 14th International Conference on Conceptual Structures (ICCS 2006), Aalborg, Denmark, pp. 144–157. Springer, Heidelberg (2006)
4. Bringmann, B., Zimmermann, A.: The chosen few: On identifying valuable patterns. In: Proceedings of the 12th IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, pp. 63–72 (2007)
5. Calders, T., Rigotti, C., Boulicaut, J.-F.: A survey on condensed representations for frequent sets. In: Boulicaut, J.-F., De Raedt, L., Mannila, H. (eds.) Constraint-Based Mining and Inductive Databases. LNCS (LNAI), vol. 3848, pp. 64–80. Springer, Heidelberg (2006)
6. Crémilleux, B., Soulet, A.: Discovering knowledge from local patterns with global constraints. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008, Part II. LNCS, vol. 5073, pp. 1242–1257. Springer, Heidelberg (2008)
7. De Raedt, L., Guns, T., Nijssen, S.: Constraint Programming for Itemset Mining. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 14th edn., Las Vegas, Nevada, USA (2008)
8. De Raedt, L.: A theory of inductive query answering. In: Proceedings of the IEEE Conference on Data Mining (ICDM 2002), Maebashi, Japan, 2002, pp. 123–130 (2002)
9. De Raedt, L., Zimmermann, A.: Constraint-based pattern set mining. In: Proceedings of the Seventh SIAM International Conference on Data Mining, Minneapolis, Minnesota, USA, April 2007, SIAM (2007)
10. ECLiPSe. Eclipse documentation, http://www.eclipse-clp.org
11. Gecode Team. Gecode: Generic constraint development environment (2006), http://www.gecode.org
12. Gervet, C.: Interval Propagation to Reason about Sets: Definition and Implementation of a Practical Language. Constraints 1(3), 191–244 (1997)
13. Giacometti, A., Miyaneh, E.K., Marcel, P., Soulet, A.: A framework for pattern-based global models. In: 10th Int. Conf. on Intelligent Data Engineering and Automated Learning, Burgos, Spain, pp. 433–440 (2009)
14. Hand, D.J.: ESF exploratory workshop on Pattern Detection and Discovery in Data Mining. In: Hand, D.J., Adams, N.M., Bolton, R.J. (eds.) Pattern Detection and Discovery. LNCS (LNAI), vol. 2447, pp. 1–12. Springer, Heidelberg (2002)
15. Kléma, J., Blachon, S., Soulet, A., Crémilleux, B., Gandrillon, O.: Constraint-based knowledge discovery from sage data. Silico Biology 8(0014) (2008)
16. Knobbe, A.: From local patterns to global models: The lego approach to data mining. In: International Workshop From Local Patterns to Global Models co-located with ECML/PKDD 2008, Antwerp, Belgium, September 2008, pp. 1–16 (2008)
17. Knobbe, A., Ho, E.: Pattern teams. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 577–584. Springer, Heidelberg (2006)
18. Lakshmanan, L.V., Ng, R., Hah, J., Pang, A.: Optimization of constrained frequent set queries with 2-variable constraints (1998)
19. Lhomme, O.: Consistency techniques for numeric csps. In: Proc. of the 13th IJCAI, Chambery, France, pp. 232–238 (1993)
20. Mamoulis, N., Stergiou, K.: Algorithms for quantified constraint satisfaction problems. In: Wallace, M. (ed.) CP 2004. LNCS, vol. 3258, pp. 752–756. Springer, Heidelberg (2004)

21. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery 1(3), 241–258 (1997)
22. Moore, R.E.: Interval analysis. Prentice-Hall, Englewood Cliffs (1966)
23. Ng, R.T., Lakshmanan, V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. In: Proceedings of ACM SIGMOD 1998, pp. 13–24. ACM Press, New York (1998)
24. Nijssen, S., Guns, T., De Raedt, L.: Correlated itemset mining in roc space: a constraint programming approach. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009), Paris, France, June 2009, pp. 647–655 (2009)
25. Siebes, A., Vreeken, J., van Leeuwen, M.: Item sets that compress. In: Proceedings of the Sixth SIAM International Conference on Data Mining, Bethesda, MD, USA, April 2006, SIAM, Philadelphia (2006)
26. Soulet, A., Crémilleux, B.: An efficient framework for mining flexible constraints. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 661–671. Springer, Heidelberg (2005)
27. Soulet, A., Klema, J., Crémilleux, B.: Efficient Mining Under Rich Constraints Derived from Various Datasets. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 223–239. Springer, Heidelberg (2007)
28. Suzuki, E.: Undirected Discovery of Interesting Exception Rules. International Journal of Pattern Recognition and Artificial Intelligence 16(8), 1065–1086 (2002)
29. Thornary, V., Gensel, J., Sherpa, P.: An hybrid representation for set constraint satisfaction problems. In: Workshop on Set Constraints co-located with the fourth Int. Conf. on Principles and Practice of Constraint Programming, Pisa, Italy (1998)
30. Yin, X., Han, J.: CPAR: classification based on predictive association rules. In: proceedings of the 2003 SIAM Int. Conf. on Data Mining (SDM 2003), San Fransisco, CA (May 2003)

Experience in Developing an Open Source Scalable Software Infrastructure in Japan

Akira Nishida

Research Institute for Information Technology, Kyushu University,
6-10-1, Hakozaki, Higashi-ku, Fukuoka, 812-8581 Japan

Abstract. The Scalable Software Infrastructure for Scientific Computing (SSI) Project was initiated in November 2002, as a five year national project in Japan, for the purpose of constructing a scalable software infrastructure to replace the existing implementations of parallel algorithms in individual scientific fields. The project covered the following four areas: iterative solvers for linear systems, fast integral transforms, their effective implementation for high performance computers of various types, and joint studies with institutes and computer vendors, in order to evaluate the developed libraries for advanced computing environments.

An object-oriented programming model was adopted to enable users to write their parallel codes by just combining elementary mathematical operations. Implemented algorithms are selected from the viewpoint of scalability on massively parallel computing environments. The libraries are freely available via the Internet, and intended to be improved by the feedback from users. Since the first announcement in September 2005, the codes have been downloaded and evaluated by thousands of users at more than 140 organizations around the world.

1 Overview

To construct a software infrastructure for highly parallel computing environments, we must precisely predict future hardware technologies, and design scalable and portable software for these technologies.

The Scalable Software Infrastructure for Scientific Computing (SSI) Project was initiated in November 2002, as a five year national project in Japan, for the purpose of constructing a scalable software infrastructure to replace the existing implementations of parallel algorithms in individual scientific fields [1,2].

Based on the above policies, we have installed various types of parallel computers, and carefully designed our libraries on them, to maintain portability and usability. The installed architectures include a shared-memory parallel computer (SGI Altix 3700), a distributed-memory parallel computer (Cray XT3), a Linux-based PC cluster, and a personal vector computer (NEC SX-6i). Since 2003, we have signed contracts with the IBM T. J. Watson Research Center on the joint study of library implementation on massively parallel environments with tens of thousands of processors. Since 2006, the SSI project has been selected for joint research with the Earth Simulator Center to port our libraries on massively

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 448–462, 2010.

parallel vector computing environments. The results of the SSI project will be evaluated on larger computers in the near future, such as the next generation supercomputer currently developed by RIKEN.

In the SSI project, we have studied object-oriented implementation of libraries [3,4,5,6], autotuning mechanisms [7], and languages for the implemented libraries [8,9]. The results are applied to the object-oriented interface of the iterative solver library Lis [10,11,12] the fast Fourier transform library FFTSS [13,14,15]. The libraries are written in C and equipped with the Fortran interfaces. In addition, we have developed SILC [16,17,18,19,20], a simple interface for library collections, to be used in parallel environments, patents of which are in application for the specifications and the extension of SILC to a scripting language.

2 Iterative Solvers

In the fields such as fluid dynamics and structural analysis, we must solve large-scale systems of linear equations to compute numerical solutions of partial differential equations, and the demand for efficient algorithms is great. The subgroup for the iterative solvers developed and released Lis, a library of iterative solvers and preconditioners with various sparse matrix storage formats [2]. Supported solvers, preconditioners, and matrix storage formats are listed in Table 1 and 2.

Table 1. Supported solvers for linear equations and eigenproblems

CG	CR
BiCG	BiCR [21]
CGS	CRS [22]
BiCGSTAB	BiCRSTAB [22]
GPBiCG	GPBiCR [22]
BiCGSafe [23]	BiCRSafe [24]
BiCGSTAB(l)	TFQMR
Jacobi	Orthomin(m)
Gauss-Seidel	GMRES(m)
SOR	FGMRES(m) [25]
IDR(s) [26]	MINRES [27]

Power Iteration
Inverse Iteration
Approximate Inverse Iteration
Conjugate Gradient [28,29]
Lanczos Iteration
Subspace Iteration
Conjugate Residual [30]

We present an example of the program using Lis in Figure 1.

There are a variety of portable software packages that are applicable to the iterative solver of sparse linear systems. SPARSKIT [38] is a toolkit for sparse matrix computations written in Fortran. PETSc [5] is a C library for the numerical solution of partial differential equations and related problems, and can be used in application programs written in C, C++, and Fortran. PETSc includes parallel implementations of iterative solvers and preconditioners based on MPI. Aztec [3] is another library of parallel iterative solvers and preconditioners and is written in C. The library is fully parallelized using MPI and can be used in

Table 2. Supported preconditioners and matrix storage formats

Jacobi	SSOR
ILU(k)	ILUT [31, 32]
Crout ILU [33, 32]	I+S [34]
SA-AMG [35]	hybrid [36]
SAINV [37]	additive Schwarz
User defined	

Compressed Row Storage	(CRS)
Compressed Column Storage	(CCS)
Modified Compressed Sparse Row	(MSR)
Diagonal	(DIA)
Ellpack-Itpack generalized diagonal	(ELL)
Jagged Diagonal	(JDS)
Block Sparse Row	(BSR)
Block Sparse Column	(BSC)
Variable Block Row	(VBR)
Dense	(DNS)
Coordinate	(COO)

```
LIS_MATRIX A;
LIS_VECTOR b,x;
LIS_SOLVER solver;
int iter;
double times;

lis_initialize(&argc,&argv);
lis_matrix_create(LIS_COMM_WORLD,&A);
lis_vector_create(LIS_COMM_WORLD,&b);
lis_vector_create(LIS_COMM_WORLD,&x);
lis_solver_create(&solver);
lis_input(A,b,x,argv[1]);
lis_vector_set_all(1.0,b);
lis_vector_duplicate(A,&x);
lis_solver_set_optionC(solver);
lis_solve(A,b,x,solver);
lis_solver_get_iters(solver,&iter);
lis_solver_get_time(solver,&times);
printf("iter = %d time = %e\n",iter,times);
lis_finalize();
```

Fig. 1. Example of the C program using Lis

applications written in C and Fortran. From the viewpoint of functionality, our library and all three of the libraries mentioned above support different sets of matrix storage formats, iterative solvers, and preconditioners. Moreover, our library is parallelized using OpenMP and takes the multicore architecture into consideration.

Feedbacks from the users have been applied to Lis, and Lis has been tested on various platforms from small PC clusters to massively parallel computers, such as NEC SX, IBM Blue Gene, and Cray XT series. The code of Lis 1.1.2 has attained the vectorization ratio of 99.1% and the parallelization ratio of 99.99%. We show a comparison of the MPI version of Lis and PETSc in Figure 2, for solving a

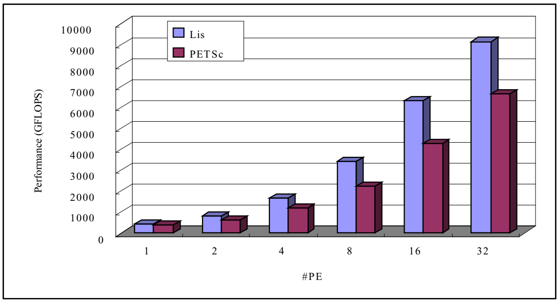


Fig. 2. Comparison of the MPI version of Lis and PETSc

three-dimensional Poisson equation (size: one million, number of nonzero entries: 26,207,180) on an SGI Altix with 32 processors.

In our project, we have designed and implemented scalable and robust algorithms of iterative solvers for linear equations and their preconditioning, derived from physical applications.

In recent years, multilevel algorithms for large-scale linear equations, such as the algebraic multigrid (AMG), have been investigated by many researches. In most cases, multigrid methods show linear scalability, and the number of iteration counts is $\mathrm{O}(n)$ for a problem of size n. The algebraic multigrid method is based on a principle similar to the geometric multigrid, which utilizes the spatial information on physical problems, but this method differs from the geometric multigrid by considering the coefficient as a vertex-edge incidence matrix. In addition, by using only the information on the elements and their relations, this method generates coarser level matrices without higher frequency errors. The complexity of the algebraic multigrid is equivalent to the geometric multigrid and can be applied to irregular or anisotropic problems. A conceptual image of the algebraic multigrid is shown in Figure 3.

We proposed an efficient parallel implementation of the algebraic multigrid preconditioned conjugate gradient method based on the smoothed aggregation (SA-AMGCG) and found that the proposed implementation provides the best performance as the problem size becomes larger [35].

Currently, the algebraic multigrid is the most effective algorithm for the general-purpose preconditioning, and its scalability is also remarkable. We have implemented the algebraic multigrid in Lis and have tested the algebraic multigrid in massively parallel environments. We presented weak scaling results for a two-dimensional Poisson equation of dimension 49 million on 1,024 nodes of a Blue Gene system in Figure 4.

In materials science, such as the solid-state physics and the quantum chemistry, large-scale simulations derived from density functional theory and first-principles calculation are often required. In these fields, there is a strong demand for efficient algorithms to solve large-scale eigenproblems. Cooperation

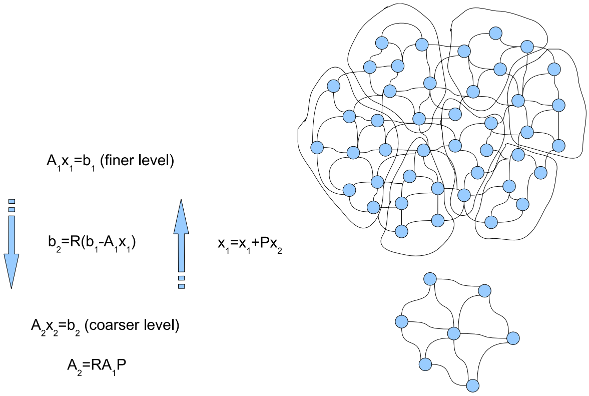


Fig. 3. Conceptual image of the SA-AMG method

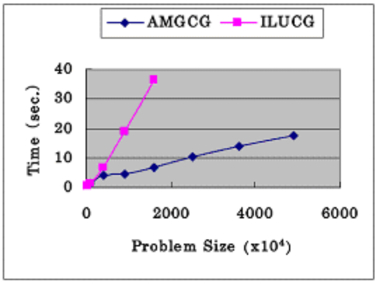


Fig. 4. Comparison of AMGCG and ILUCG (for linear equations)

with such fields is desirable in order to develop scalable eigensolvers. There are several methods to compute eigenvalues of large-scale sparse matrices, including the Lanczos method for symmetric problems, the Arnoldi method, its extension for nonsymmetric problems, the Davidson method originally proposed for quantum chemistry, and the Jacobi-Davidson method, a derivative of the Davidson method. Based on observations, we proposed that the scalability of the conjugate gradient method for linear equations can improve the performance of eigensolvers in parallel environments, where the extreme eigenvalues of a generalized eigenproblem can be solved by reducing these problems to the calculation of the local maximum or local minimum of the Rayleigh quotients combined with appropriate preconditioners, such as the algebraic multigrid [39,29]. In the fiscal year 2008, we have focused on the implementation of the existing major eigensolvers for sparse matrices on Lis, which was released as version 1.2.0 of Lis in 2008. Performance evaluation shown in Figure 4, indicates the scalability of our implementation and the advantage of the conjugate gradient method.

3 Fast Integral Transforms

In the fields such as the hydrodynamics and the weather forecasting, we need to solve problems on a spherical surface, which derives the demands for high-performance fast integral transforms. This subgroup has developed scalable fast integral transform libraries [13,14,15], which have practical performance in real computing environments.

The fast Fourier transform is an implementation of the discrete Fourier transform and is used in many fields, ranging from large-scale scientific computing to image processing. Although many improvements have been proposed since the discovery of the FFT algorithm [40], recent rapid progress in processor architectures requires new FFT kernels.

The existing algorithms, which propose efficient use of cache memory, adopt algorithms that do not require bit-reverse, such as Stockham FFT. In the SSI project, we overlapped the bit-reverse process with memory access to enable an in-place algorithm, and we have shown that the latency can be eliminated.

Processors like Intel's IA-64 and IBM's POWER, which have two multiply-add units and operate four floating point calculations per cycle, are becoming the mainstream. The multiply-add operation is a combination of multiply and add, while a single multiply or a single add operation also uses the multiply-add unit. This implies that we must combine as many multiplies and adds as possible in order to utilize the units efficiently [41,42,43,44]. We proposed an 8-radix FFT kernel with the least number of multiply-add operations, which requires a smaller twiddle factor table and a smaller number of twiddle factors to be loaded. The result is reflected in the FFTSS library, which we developed as an FFT library for superscalar processors with automatic performance tuning mechanism, as shown in Figure 5.

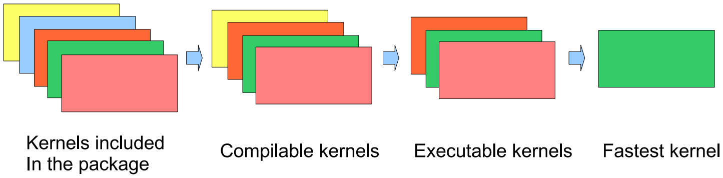


Fig. 5. Automatic performance tuning mechanism of FFTSS

A program example and the performance of FFTSS for one-dimensional FFT, as compared with commercial libraries, and for OpenMP-based two-dimensional parallel FFT supporting padding, as compared with FFTW, are shown in Figure 6 and 7-9, respectively.

We have also developed an MPI version of FFTSS and implemented a vector processor version of FFT, overlapping a huge number of all-to-all communications and computation, as part of a joint study with the Earth Simulator

```
max_threads = omp_get_num_procs();
fftss_plan_with_nthreads(max_threads);
plan = fftss_plan_dft_2d(nx, ny, py, vin, vout,
FFTSS_FORWARD, FFTSS_MEASURE);
{ /* Initialization of array */ }
for (nthreads = 1;
 nthreads <= max_threads; nthreads ++)
{
fftss_plan_with_nthreads(nthreads);
t = fftss_get_wtime();
fftss_execute(plan);
t = fftss_get_wtime() - t;
printf("%1f sec. with %d threads.\n",
 nthreads, t);
}
```

Fig. 6. Program example of FFTSS

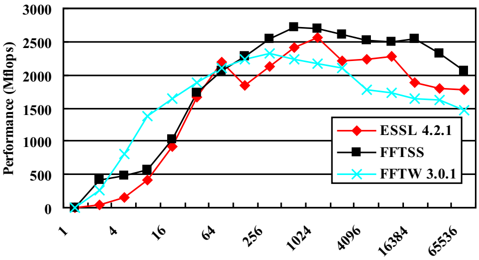


Fig. 7. Performance comparison of 1-dimensional FFT kernels on IBM POWER5

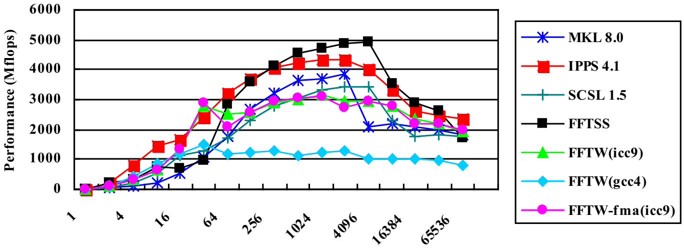


Fig. 8. Performance comparison of 1-dimensional FFT kernels on Intel Itanium2

Center. Our library showed the best performance of 16.3 TFLOPS with the double-precision FFT on 512 nodes of the Earth Simulator, which is 49.6% of the peak.

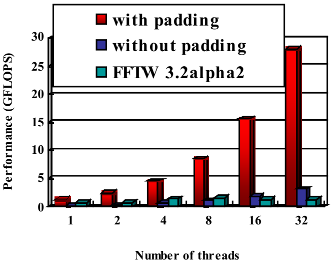


Fig. 9. Performance of OpenMP-based two-dimensional parallel FFTSS on SGI Altix 3700

4 Programming Environment

Libraries for matrix computation are indispensable to scientific computations, and several libraries have been proposed for their implementation. These libraries are provided with APIs to be used with other programs. For example, to solve a linear equation Ax = b, the user prepares a matrix A and a vector b in the format specified by the library and calls a function with specified arguments. In such cases, the program created by the user depends on the data structure and function calls of the specific library. In many cases, there are no compatibilities between the interfaces of the libraries, and the user must modify the program to use the routines provided by other libraries. These libraries are also applicable to cases with different preconditioners or different computing precisions. There are also libraries for specific computing environments, which require libraries to be changed and codes to be rewritten. It is burdensome for the user to rewrite programs, and a more flexible method of using libraries is needed. To fulfill the demand, we have proposed an environment-independent matrix computation library SILC, a simple interface for library collections [16,17,18,19,20].

Apart from the former usage based on the specific interface of a library, SILC utilizes the features of the matrix computation libraries by sending three types of requests: (1) deposit of data to be input, (2) requests for computation by means of mathematical expressions in the form of text, and (3) fetch data to be output. The input data, such as matrices and vectors, are transferred to an independent memory space from the user program. The requests of computation by means of mathematical expressions are interpreted as appropriate function calls and are executed in the independent memory space. Finally, the results are returned to the memory space of the user program by request, as shown in Figure 10.

As an example, we present a C program in Fig. 13, which calls a routine of LAPACK to solve a linear equation via the interface of SILC.

After making matrix A and vector b in LAPACK's format, this program calls the solver routine of LAPACK via the three routines SILC_PUT, SILC_EXEC, and SILC_GET provided by SILC. For scientific computing, libraries based on OpenMP and MPI are used in various parallel computing environments. SILC

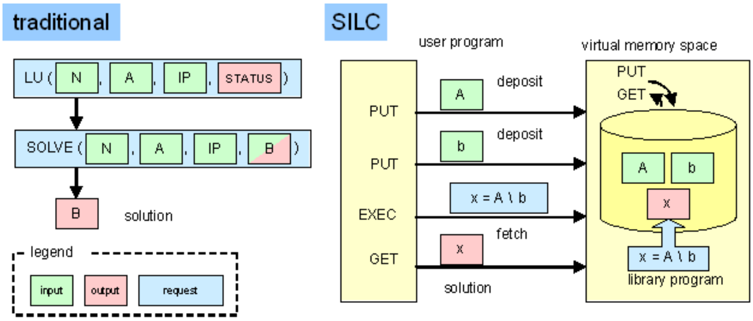


Fig. 10. Concept of SILC, a simple interface for library collections

```
silc_envelope_t A, b, x;
/* make matrix A and vector b */
SILC_PUT("A", &A);
SILC_PUT("b", &b);
/* solve the linear equation */
SILC_EXEC("x = A \\ b");
SILC_GET(&x, "x");
```

Fig. 11. C program calling a routine of LAPACK to solve a linear equation via the interface of SILC

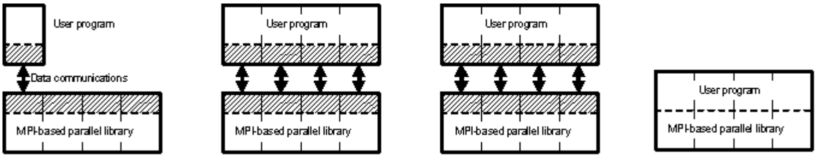

User program
Data communications
MPI-based parallel library
User program
MPI-based parallel library
User program
MPI-based parallel library
User program
MPI-based parallel library

Fig. 12. System confugrations of SILC

buffers the difference of computing environments between the user program and computing environments and enables us to use the libraries in a language- and environment-independent manner. We have assumed the following four situations: (A) sequential client and sequential server (B) sequential client and shared-memory parallel server (C) sequential client and distributed-memory parallel server (D) distributed-memory parallel client and distributed-memory parallel server The structure of the user programs are shown in Figure 12.

Libraries are used by linking to the user program. The user program transfers data and requests computations by connecting to a single process or multiple processes. The data transferred from the user program to the server is transformed into the requested data distribution manner and is retained in the server

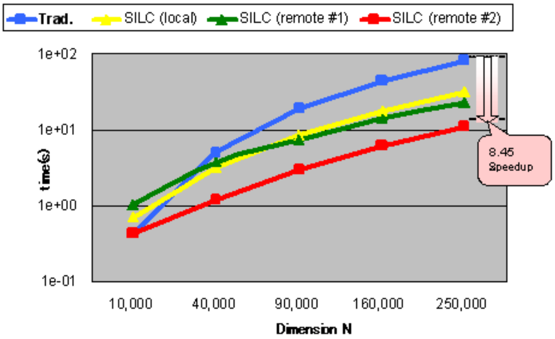


Fig. 13. Performance of SILC on distributed-memory parallel computing environments

Table 3. Configuration of computing environments

	User Program	SILC Server
traditional	Xeon4 (1PE)	
SILC(local)	Xeon4 (1PE)	Xeon4 (4PEs)
SILC(remote #1)	Xeon4 (1PE)	Xeon4 (8PEs)
SILC(remote #2)	Xeon4 (1PE)	Xeon4 (16PEs)

Table 4. Specification of machine architectures

Host	Specification
Xeon4	Intel Xeon 2.8GHz x2, 1GB RAMM, Red Hat Linux 8.0, LAM/MPI 7.0
Xeon8	Eight different nodes in the same cluster as Xeon4
Altix	Intel Itanium2 1.3GHz x32, 32GB RAMM, Red Hat Linux AS 2.1, SGI MPI 4.4

processes. The results returned to the user program are transformed again to the requested data distribution manner by the data redistribution mechanism.

Using remote distributed-memory parallel computing environments via the implemented system, we have observed better performance compared with the user program written in the traditional manner. The results of the solution of the initial value problem of a two-dimensional diffusion equation are shown using the finite difference method shown in Figure 13. Configuration is shown in Table 3-4. We have used the conjugate gradient method without a preconditioner of Lis for the solution of linear equations, and have interpreted the request to solve the system into the function call of MPI-based Lis. Denoting the dimension by N and the number of iterations by I, the amount of communication is $O(N)$ and the complexity is $O(NI)$, which shows that we can solve the problems faster on the remote parallel server than on the local client when several iterations are required, even considering the communication cost.

To use control statements such as conditional branches and loops in SILC, the user must prepare a function, which includes the statements, and must make it callable from the library program. This makes it impossible to program arbitrary combinations of mathematical expressions and control statements. In the SSI project, we extended SILC to a scripting language with control statements, which interprets the user program and separates the mathematical expressions and the control statements and processes them using the existing SILC framework. By using this extension, we are able to write programs in simple manner.

5 Conclusion

We have overviewed our experience of the SSI project, one of the first national projects for open source general purpose scalable numerical libraries which originally intended to catch up with the global standard of freely distributed scientific software, and contribute to the development of computational science. It is just the first step for us to achieve more flexibility in scalable scientific computing, but we hope our efforts reduce some barriers towards upcoming massively parallel scientific computing environments in the near future.

Acknowledgements

The author is grateful for Dr. Hisashi Kotakemori, Dr. Akihiro Fujii, Dr. Akira Nukada, Dr. Yuichiro Hourai, Dr. Tamito Kajiyama, Prof. Hidehiko Hasegawa, Prof. Reiji Suda, Prof. Daisuke Takahashi, Prof. Shao-Liang Zhang, Prof. Kengo Nakajima, Prof. Kuniyoshi Abe, Prof. Shoji Itoh, Prof. Tomohiro Sogabe, and Miss Kaori Sato, for their long support as the project members, and for the support from many people and organizations. This study was partially supported by Department of Computer Science, the University of Tokyo, the CREST program, Japan Science and Technology Agency, NIWS Corporation, T. J. Watson Research Center, IBM Corporation, the Earth Simulator Center, JAMSTEC, Research Institute for Information Technology, Kyushu University, and Cyberscience Center, Tohoku University.

References

1. Nishida, A.: SSI: Overview of simulation software infrastructure for large scale scientific applications (in Japanese). Technical Report 2004-HPC-098, IPSJ (2004)
2. Nishida, A., Suda, R., Hasegawa, H., Nakajima, K., Takahashi, D., Kotakemori, H., Kajiyama, T., Nukada, A., Fujii, A., Hourai, Y., Zhang, S.L., Abe, K., Itoh, S., Sogabe, T.: The Scalable Software Infrastructure for Scientific Computing Project. Kyushu University (2009), http://www.ssisc.org/
3. Tuminaro, R.S., Heroux, M., Hutchinson, S.A., Shadid, J.N.: Official Aztec User's Guide, Version 2.1. Technical Report SAND99-8801J, Sandia National Laboratories (1999)

4. Wu, K., Milne, B.: A survey of packages for large linear systems. Technical Report LBNL-45446, Lawrence Berkeley National Laboratory (2000)
5. Balay, S., Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc Users Manual. Technical Report ANL-95/11, Argonne National Laboratory (2004)
6. Dongarra, J., Ltaief, H.: Freely available software for linear algebra on the Web (2009), http://www.netlib.org/utk/people/JackDongarra/la-sw.html
7. Frigo, M., Johnson, S.G.: The design and implementation of FFTW3. Proceedings of the IEEE 93(2), 216–231 (2005)
8. Moler, C.: Design of an interactive matrix calculator. In: AFIPS National Computer Conference. AFIPS Conference Proceedings, vol. 49, pp. 363–368. AFIPS Press (1980)
9. Kennedy, K., Broom, B., Chauhan, A., Fowler, R., Garvin, J., Koelbel, C., McCosh, C., Mellor-Crummey, J.: Telescoping Languages: A System for Automatic Generation of Domain Languages. Proceedings of the IEEE 93, 387–408 (2005)
10. Kotakemori, H., Hasegawa, H., Nishida, A.: Performance Evaluation of a Parallel Iterative Method Library using OpenMP. In: Proceedings of the 8th International Conference on High Performance Computing in Asia Pacific Region, pp. 432–436 (2005)
11. Kotakemori, H., Hasegawa, H., Kajiyama, T., Nukada, A., Suda, R., Nishida, A.: Performance evaluation of parallel sparse matrix-vector products on SGI Altix3700. In: Mueller, M.S., Chapman, B.M., de Supinski, B.R., Malony, A.D., Voss, M. (eds.) IWOMP 2005 and IWOMP 2006. LNCS, vol. 4315, pp. 153–163. Springer, Heidelberg (2008)
12. Kotakemori, H., Fujii, A., Hasegawa, H., Nishida, A.: Implementation of Fast Quad Precision Operation and Acceleration with SSE2 for Literative Solver Library (in Japanese). IPSJ Transactions on Advanced Computing Systems 1(1), 73–84 (2008)
13. Nukada, A.: FFTSS: a High Performance Fast Fourier Transform Library. In: Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. III, pp. 980–983. IEEE Computer Society Press, Washington (2006)
14. Nukada, A., Takahashi, D., Suda, R., Nishida, A.: High Performance FFT on SGI Altix 3700. In: Perrott, R., Chapman, B.M., Subhlok, J., de Mello, R.F., Yang, L.T. (eds.) HPCC 2007. LNCS, vol. 4782, pp. 396–407. Springer, Heidelberg (2007)
15. Nukada, A., Hourai, Y., Nishida, A., Akiyama, Y.: High Performance 3D Convolution for Protein Docking on IBM Blue Gene. In: Stojmenovic, I., Thulasiram, R.K., Yang, L.T., Jia, W., Guo, M., de Mello, R.F. (eds.) ISPA 2007. LNCS, vol. 4742, pp. 958–969. Springer, Heidelberg (2007)
16. Kajiyama, T., Nukada, A., Hasegawa, H., Suda, R., Nishida, A.: LAPACK in SILC: Use of a Flexible Application Framework for Matrix Computation Libraries. In: Proceedings of the 8th International Conference on High Performance Computing in Asia Pacific Region, pp. 205–212. IEEE, Washington (2005)
17. Kajiyama, T., Nukada, A., Hasegawa, H., Suda, R., Nishida, A.: SILC: A Flexible and Environment Independent Interface for Matrix Computation Libraries. In: Wyrzykowski, R., Dongarra, J., Meyer, N., Waśniewski, J. (eds.) PPAM 2005. LNCS, vol. 3911, pp. 928–935. Springer, Heidelberg (2006)
18. Kajiyama, T., Nukada, A., Suda, R., Hasegawa, H., Nishida, A.: A Performance Evaluation Model for the SILC Matrix Computation Framework. In: Proceedings of the IFIP International Conference on Network and Parallel Computing, pp. 93–103. The University of Tokyo, Tokyo (2006)

19. Kajiyama, T., Nukada, A., Suda, R., Hasegawa, H., Nishida, A.: Distributed SILC: An Easy-to-Use Interface for MPI-based Parallel Matrix Computation Libraries. In: Kågström, B., Elmroth, E., Dongarra, J., Waśniewski, J. (eds.) PARA 2006. LNCS, vol. 4699, pp. 860–870. Springer, Heidelberg (2007)
20. Kajiyama, T., Nukada, A., Suda, R., Hasegawa, H., Nishida, A.: Cloth simulation in the SILC matrix computation framework: A case study. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2007. LNCS, vol. 4967, pp. 1086–1095. Springer, Heidelberg (2008)
21. Sogabe, T., Sugihara, M., Zhang, S.: An Extension of the Conjugate Residual Method for Solving Nonsymmetric Linear Systems(in Japanese). Transactions of the Japan Society for Industrial and Applied Mathematics 15(3), 445–460 (2005)
22. Abe, K., Sogabe, T., Fujino, S., Zhang, S.: A Product-type Krylov Subspace Method Based on Conjugate Residual Method for Nonsymmetric Coefficient Matrices (in Japanese). IPSJ Transactions on Advanced Computing Systems 48(SIG8(ACS18)), 11–21 (2007)
23. Fujino, S., Fujiwara, M., Yoshida, M.: BiCGSafe method based on minimization of associate residual (in Japanese). Transactions of JSCES 8(20050028), 145–152 (2005), http://save.k.u-tokyo.ac.jp/jsces/trans/trans2005/No20050028.pdf
24. Fujino, S., Onoue, Y.: Estimation of BiCRSafe method based on residual of BiCR method (in Japanese). Technical Report 2007-HPC-111, IPSJ (2007)
25. Saad, Y.: A Flexible Inner-outer Preconditioned GMRES Algorithm. SIAM J. Sci. Stat. Comput. 14, 461–469 (1993)
26. Soonerveld, P., van Gijzen, M.B.: IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations. SIAM J. Sci. Comput. 31, 1035–1062 (2008)
27. Greenbaum, A.: Iterative Methods for Solving Linear Systems. SIAM, Philadelphia (1997)
28. Knyazev, A.V.: Toward the Optimal Preconditioned Eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method. SIAM J. Sci. Comput. 23(2), 517–541 (2001)
29. Nishida, A.: A Short Survey of Applications and Evaluations of Preconditioned Conjugate Gradient Method for Large Scale Eigenvalue Problems (in Japanese). In: Proceedings of the 2003 Annual Conference, JSIAM, Tokyo, pp. 326–327 (2003)
30. Suetomi, E., Sekimoto, H.: Conjugate gradient like methods and their application to eigenvalue problems for neutron diffusion equation. Ann. Nucl. Energy 18(4), 205–227 (1991)
31. Saad, Y.: ILUT: a dual threshold incomplete *LU* factorization. Numerical linear algebra with applications 1(4), 387–402 (1994)
32. Li, N., Suchomel, B., Osei-Kuffuor, D., Saad, Y.: ITSOL: ITERATIVE SOLVERS package. In: University of Minnesota (2008), http://www-users.cs.umn.edu/~saad/software/ITSOL/
33. Li, N., Saad, Y., Chow, E.: Crout version of ILU for general sparse matrices. SIAM J. Sci. Comput. 25, 716–728 (2003)
34. Kohno, T., Kotakemori, H., Niki, H.: Improving the Modified Gauss-Seidel Method for Z-matrices. Linear Algebra and its Applications 267, 113–123 (1997)
35. Fujii, A., Nishida, A., Oyanagi, Y.: Evaluation of Parallel Aggregate Creation Orders: Smoothed Aggregation Algebraic Multigrid Method, pp. 99–122. Springer, Berlin (2005)
36. Abe, K., Zhang, S., Hasegawa, H., Himeno, R.: A SOR-base Variable Preconditioned CGR Method (in Japanese). Trans. JSIAM 11(4), 157–170 (2001)

37. Bridson, R., Tang, W.P.: Refining an approximate inverse. J. Comput. Appl. Math. 123, 293–306 (2000)
38. Saad, Y.: SPARSKIT: a basic tool kit for sparse matrix computations, version 2. University of Minnesota (1994), http://www.cs.umn.edu/saad/software/
39. Nishida, A., Oyanagi, Y.: Performance Evaluation of Low Level Multithreaded BLAS Kernels on Intel Processor based cc-NUMA Systems. In: Veidenbaum, A., Joe, K., Amano, H., Aiso, H. (eds.) ISHPC 2003. LNCS, vol. 2858, pp. 500–510. Springer, Heidelberg (2003)
40. Duhamel, P., Hollmann, H.: Split-Radix FFT Algorithm. Electron. Lett. 20, 14–16 (1984)
41. Linzer, E.N., Feig, E.: Implementation of Efficient FFT Algorithms on Fused Multiply-Add Architectures. IEEE Trans. Signal Processing 41, 93–107 (1993)
42. Goedecker, S.: Fast Radix 2,3,4 and 5 Kernels for Fast Fourier Transformations on Computers with Overlapping Multiply-Add Instructions. SIAM J. Sci. Comput. 18, 1605–1611 (1997)
43. Karner, H., Auer, M., Ueberhuber, C.W.: Multiply-Add Optimized FFT Kernels. Math. Models and Methods in Appl. Sci. 11, 105–117 (2001)
44. Wait, C.D.: IBM PowerPC 440 FPU with complex arithmetic extensions. IBM Journal of Research and Development 49(2/3), 249–254 (2005)
45. Bailey, D.H.: A fortran-90 double-double library. In: Lawrence Berkeley National Laboratory (2008), http://www.nersc.gov/dhbailey/mpdist/mpdist.html
46. Hida, Y., Li, X.S., Bailey, D.H.: Algorithms for quad-double precision floating point arithmetic. In: Proceedings of the 15th Symposium on Computer Arithmetic, pp. 155–162. IEEE, Washington (2001)
47. Dekker, T.: A floating-point technique for extending the available precision. Numerische Mathematik 18, 224–242 (1971)
48. Knuth, D.E.: The Art of Computer Programming: Seminumerical Algorithms, vol. 2. Addison-Wesley, New Jersey (1969)
49. Bailey, D.H.: High-Precision Floating-Point Arithmetic in Scientific Computation. Computing in Science and Engineering 7, 54–61 (2005)
50. Intel: Intel Fortran Compiler User's Guide Vol I. Intel (2009)
51. Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., der Vorst, H.V.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd edn. SIAM, Philadelphia (1994)
52. Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., van der Vorst, H. (eds.): Templates for the Solution of Algebraic Eigenvalue Problems. SIAM, Philadelphia (2000)
53. Lehoucq, R.B., Sorensen, D.C., Yang, C.: ARPACK Users' Guide: Solution of Large-scale Eigenvalue Problems with implicitly-restarted Arnoldi Methods. SIAM, Philadelphia (1998)
54. Bramley, R., Wang, X.: SPLIB: A library of iterative methods for sparse linear system. Technical report, Indiana University–Bloomington (1995)
55. Boisvert, R.F., Pozo, R., Remington, K., Barrett, R., Dongarra, J.J.: The Matrix Market: A web resource for test matrix collections, pp. 125–137. Chapman & Hall, London (1997)
56. Casanova, H., Dongarra, J.: NetSolve: A Network Server for Solving Computational Science Problems. In: The International Journal of Supercomputer Applications and High Performance Computing, pp. 212–223. MIT Press, Boston (1995)
57. Sato, M., Nakada, H., Sekiguchi, S., Matsuoka, S., Nagashima, U., Takagi, H.: Ninf: A network based information library for global world-wide computing infrastructure (1997)

58. Rose, L.D., Padua, D.: Techniques for the translation of MATLAB programs into Fortran 90. ACM Transactions on Programming Languages and Systems 21, 286–323 (1999)
59. Kawabata, H., Suzuki, M., Kitamura, T.: A MATLAB-based code generator for sparse matrix computations. In: Chin, W.-N. (ed.) APLAS 2004. LNCS, vol. 3302, pp. 280–295. Springer, Heidelberg (2004)
60. MathWorks, I.: Matlab. MathWorks, Inc (2005), http://www.mathworks.com/
61. Luszczek, P., Dongarra, J.: Design of interactive environment for numerically intensive parallel linear algebra calculations. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2004. LNCS, vol. 3039, pp. 270–277. Springer, Heidelberg (2004)
62. Fujii, A., Suda, R., Nishida, A., Oyanagi, Y.: Evaluation of Asynchronous Iterative Method for Sparse Matrix Solver. In: Proceedings of the Second international Workshop on Automatic Performance Tuning, pp. 43–51. The University of Tokyo, Tokyo (2007)
63. Kajiyama, T., Nukada, A., Suda, R., Hasegawa, H., Nishida, A.: Toward Automatic Performance Tuning for Numerical Simulations in the SILC Matrix Computation Framework. In: Proceedings of the Second international Workshop on Automatic Performance Tuning, pp. 81–90. The University of Tokyo, Tokyo (2007)
64. Nishida, A.: Building Cost Effective High Performance Computing Environment via PCI Express. In: Proceedings of the 2006 International Conference on Parallel Processing Workshops, pp. 519–526. IEEE, Washington (2006)
65. Fujii, A., Suda, R., Nishida, A.: Parallel Matrix Distribution Library for Sparse Matrix Solvers. In: Proceedings of the 8th International Conference on High Performance Computing in Asia Pacific Region, pp. 213–219. IEEE, Washington (2005)
66. Hourai, Y., Nishida, A., Oyanagi, Y.: Network-aware Data Mapping on Parallel Molecular Dynamics. In: Proceedings of 11th International Conference on Parallel and Distributed Systems, pp. 126–132. IEEE, Washington (2005)

A New Formal Test Method for Networked Software Integration Testing

Shuai Wang[1,2], Yindong Ji[1,2], Wei Dong[2], and Shiyuan Yang[1]

[1] Department of Automation, Tsinghua University
[2] Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing, China
wangshuai81@gmail.com

Abstract. This paper considers the integration testing for networked software that is built by assembling several distributed components in an interoperable manner. Using the traditional single automata-based test approaches, we suffer from the state combinatorial explosion problem. Moreover, several generated test cases may not be executable. This paper proposed a test method based on the automata net which is the extension of communication automata. The state/transition path (S/T-Path) is defined to describe the execution of the software under test. The test cases are constructed through combining the atomic S/T-Paths and all executable. The test cases are calculated from the local transition structures and the interaction procedure between components, so the state combinatorial explosion problem will not be encountered. The generation of test cases for certain software and the benefits for the problems are discussed. Results show that our method has better properties.

Keywords: Networked software; integration testing; automata net; test coverage rule; State/Transition-Path.

1 Introduction

Networked software is a type of distributed software system that is built by assembling several components in an interoperable manner. These components are distributed in different computers and connected through network.

Testing is an important work for the validity and reliability of networked software, which can be studied in different levels and form different points of view. S. Ghosh and his colleagues have done some researches on the networked software testing. They proposed a test method based on interfaces [1]. I.-H. Cho discussed that the aim of integration testing for networked software is to detect faults in the interfaces among components [2]. Despite several researches have been done for integration testing of networked software. Most of these approaches are informal and some discussions focus on random testing. So by using these methods, it is difficult to implement test automatically and systematically.

The presence of a formal model or specification, which defines the required behaviors of the software, introduces the possibility of automating or semi-automating much of the testing process. This can lead to more effective and efficient testing.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 463–474, 2010.

There are many approaches to formally modeling, or specifying, a software system. Some formal methods have been used in software test [3-6]. Particularly, the formal methods based on automata model are widely studied and applied for single software [4-7]. In this paper, we extend the formal method on the networked software testing.

The networked software may be more naturally and simply modeled by a set of automata, rather than a single automaton, those operate concurrently and may interact by changing messages. Then the behaviors of the software can be described by a compound automaton which is the equivalent single automaton converted from the set of component automatons through automata product operation [8]. Tests can be generated from the compound automaton using standard automata test techniques [9-10]. Suppose that the model of certain networked software consists of automata $A_1,\cdots A_n$. Then the number of states of the compound automaton is $\prod_i |A_i|$, where $|A_i|$ means the number of states of automaton A_i; thus this approach may suffer from the state combinatorial explosion problem.

Take the model in Fig. 1 for example. We suppose that the software is composed of two components, and each is modeled as an automaton, namely A_1, A_2. The states and transitions for each automaton are shown in Fig. 1.A$_1$ and Fig. 1.A$_2$. The compound automaton get through product operation is shown in Fig. 1.A. Each component automaton contains three states, and the compound automaton contains nine states. If the software has many components or each automaton contains many states, the state number of compound automaton may be very large.

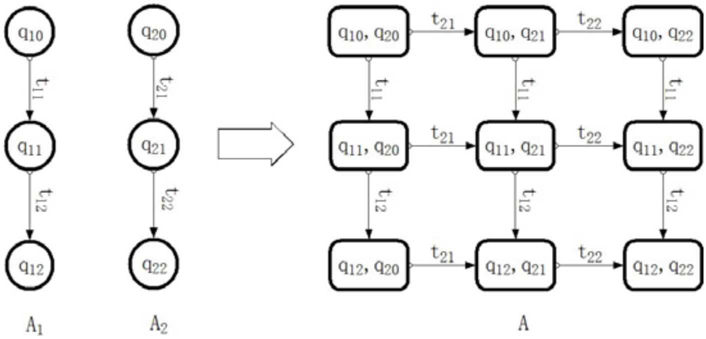


Fig. 1. Combining the automata of the software components

The main problems of integration testing for networked software with single automaton are:

1) State combinatorial explosion problem [11].
2) Some test case combining the local transition paths may be unexecutable. If the specification of software constrains that the component A_1 can trigger transition t_{12} only after it receives the output of transition t_{21}, then the transition path $t_{11}, t_{12}, t_{21}, t_{22}$ cannot be carried out. The existence of this unexecutable test

sequences are caused by the unreachable state of the compound automata when certain specification is required.

Despite some approaches to simplify the states of the compound automata and delete the unreachable state [12-13], the test implementation may be an impossible work when the networked software is very complex.

In this paper, we proposed a novel formal test method to solve these problems. By extending the communication automata, we set up the automata net model as the formal test model for networked software. Because we do not need to combine the component automata, we will not meet the state combinatorial explosion problem. Especially, the state number of this model is much smaller than the compound automata model. Testing the validity of the networked software is implemented through checking the outputs after sequences of inputs that are applied. At same time, the state/transition path (S/T-Path) is defined to describe the atomic execution of the software. The test cases are generated through combing the atomic S/T-Paths with different operators. At same time, we proposed an atomic S/T-Path coverage rule as the aim of test case generation.

The rest of this paper is organized as follows. Some formal models for software testing are discussed in Section 2. The construction approach of test cases for networked software is introduced in Section 3. The generation of test cases for certain software and its benefits for the problems that are met by single compound automaton method are discussed in 4. Finally, conclusion is presented in Section 5.

2 Formal Model

Formal model is the foundation of software testing. In this section, we will introduce the formal model for networked software.

2.1 Classical Automata Model

Behaviors of software cannot be described by differential or difference equations, but they can be modeled as automata whose behaviors are described by the traces (or sequence) of events that record significant qualitative changes in the state of software. The finite state machine is the commonly used automaton model for software testing, and it is formally defined as follows:

Definition 1. A finite state machine (FSM) is a six-tuple $FSM = (Q, q_0, \Sigma, \Lambda, \delta, \lambda)$ [14], where:

1) Q is the set of states;
2) Σ is the finite set of inputs;
3) Λ contains all outputs;
4) $\delta : Q \times \Sigma \to Q$ is the state transition function;
5) q_0 is the initial state;
6) $\lambda : Q \times \Sigma \to \Lambda$ is the output function.

A FSM can be represented by a directed graph $G = (V, E)$, where the set $V = \{v_1, \cdots v_n\}$ of vertices represents the set of specified states Q of the machine and directed edges represent transitions from one state to another in it. An edge in G is fully specified by a triple $(v_i, v_j; L)$, where $L \equiv i_k / o_l$, $L^{(i)} \equiv i_k$ and $L^{(o)} \equiv o_l$. In this paper, it is assumed that G is strongly connected.

When the software is modeled as a FSM, the testing of software can be taken as checking the output value of several sequences of input values. Several methods have been studied to test the software based on FSM [15-16].

2.2 Automata Net Model

Networked software strengths the computing capability of system, but the test and validation of software become a difficult work. The networked software has multi-thread, distributed and parallel properties. It may be more naturally and simply modeled by a set of automata rather than a single automaton. The entities of all component automata are called automaton net. In this model, each component automaton describes the behaviors of software component. The interaction behaviors between components are described by channel automata.

Definition 2. An automaton net is formally defined by a two-tuple $Anet = (A, C)$.

1) $A = \{A_1, A_2, \cdots A_n\}$ is the finite set of component automata;
2) $C = \{c_{i,j} : i, j \leq n \wedge i \neq j\}$ is the set of finite channel automata, and $c_{i,j}$ is the channel automaton between software component A_i and component A_j.

A component automaton $A_i \in A$ is a classical FSM that is defined in the definition 1. Its states are called local states and its transitions are named local transitions. These are in contrast to global transitions and global states which are defined on the compound automaton of component automata.

We define channel automaton $c_{i,j}$ to describe the message transporting behaviors between component automata. The behaviors of $c_{i,j}$ are decided by the channel properties.

Definition 3. A channel automaton is a four-tuple $c_{i,j} = (Q_c, \delta_c, q_{c,0}, M_{ij})$, where:

1) Q_c is the set of finite channel states, its number is decided by the properties and buffer limit of communication channel;
2) δ_c is the state transition function of communication channel;
3) $q_{c,0}$ is the initial state of communication channel;
4) $M_{i,j}$ is the set of finite messages transporting through channel $c_{i,j}$. $m \in M_{i,j}$ is one of the messages that are exchanged between A_i and A_j.

Since this paper is concerned with the testing of the transition to verify the correctness of software, the properties of channel will not be discussed here.

According to the properties of local transition, it can be divided into three types:

1) Non-communicating transition: the input of this type transition will be applied at the input port of this component and output can be observed at the output port. It is formally defined as $(q_i, q_j; i_k / o_l)$.

2) Sending message transition: the input of this type transition will be applied at the input port of this component, but the output will be sent to other component. It is formally defined as $(q_i, q_j; i_k / A_j\,!o_{ij})$.
3) Receiving message transition: the input of this type transition is received from output of other component, but output can be observed at output port of this component. It is defined as $(q_i, q_j; A_i\,?o_{ij} / o_k)$.

Where $A_j\,!o_{ij}$ means sending message o_{ij} to component A_j, and $A_i\,?o_{ij}$ means receiving message o_{ij} from A_i.

Example 1. Certain networked software is modeled as automata net $Anet = (A, C)$ shown in Fig. 2, where

$A = \{A_1, A_2, A_3, A_4\}; C = \{C_{12}, C_{13}, C_{23}, C_{24}, C_{34}\};$
$M_{1,2} = \{e\}; M_{1,3} = \{x\}; M_{2,3} = \{h\};$
$M_{2,4} = \{d\}; M_{3,4} = \{f\}.$

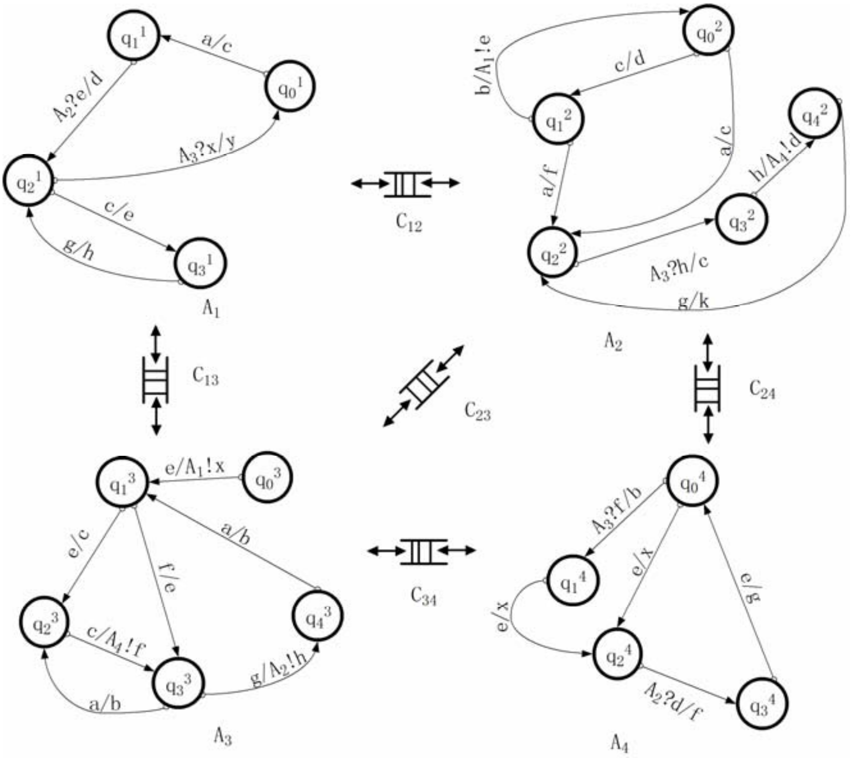


Fig. 2. An automata net model example for certain networked software

2.3 Compound Automaton

If input values will only be received in stable states, the full behavior of software is equivalent to the compound automaton [4].

Each software component is modeled as an component automaton A_i, then the compound model of the networked software is compute by

$$A = A_1 \times A_2 \times \cdots \times A_n$$

where × means the product operation on automata. We use $P(A)$ to denote the compound automaton generated from A, X and Y to denote the input and output sets of A, $S = Q_1 \times Q_2 \times \cdots \times Q_n$ to denote the set of stable global states. $S(k)$ denotes the state of the *k*th component. Clearly some elements in S may be unreachable. We use S_r to denote the reachable state in S. The initial state of $P(A)$ is $s_0 = (q_0^1, q_0^2, \cdots q_0^3)$. The state transition functions and output functions are also denoted as δ and λ. Thus $P(A)$ is defined by

$$P(A) = (S_r, s_0, X, Y, \delta, \lambda)\,. \tag{1}$$

Example 2. If we compute the compound automaton of the networked software in example 1, its state number is 400. Despite some of these states are unreachable, the reachable state number is still very large. The attempt of drawing the state transition graph is impossible. This state combinatorial explosion problem causes the traditional test approach for single FSM to be impossible for complex networked software system. In this example, $(q_0^1, q_0^2, q_0^3, q_0^4)$ is the global initial state. $(q_0^1, q_0^2, q_0^3, q_2^4)$ is a global state. $((q_0^1, q_0^2, q_0^3, q_0^4), (q_0^1, q_0^2, q_0^3, q_2^4), e/x)$ is a global transition.

3 Construction of Test Case

This section shall consider the problem of the construction of test case for networked software integration testing. As discussed in previous section, if we construct the test cases based on the compound automaton, we may meet the state combinatorial explosion problem. In order to solve this problem, we give another construction method that is defined on the local transition structures of component automata, in which the combinatorial explosion problem will not be encountered.

The construction of test cases is based on the following assumptions:

Assumption 1. We assume that message transporting time and delay time are all zero, then $s_i!m_{ij}$, $s_i?m_{ij}$, $s_j!m_{ij}$ and $s_j?m_{ij}$ are synchronous. This means that the behaviors of channel between two component automata are modeled as empty channel.

Assumption 2. The faults of two different transitions are independent. This means that the faulty output of certain transition cannot be corrected by the following transitions. So a test case is said to pass when all transitions in it generate right outputs.

3.1 Formal Description of the Execution of Network Software

The testing of networked software can be taken as checking the output value at different output ports after several sequences of input values are applied at different input ports. If the outputs are expected, the networked software is said to be right. The input sequences and required outputs can be gotten from the global transition structure of compound automata if the prevalent automaton-based test approach is applied. Since the global transition of compound automaton is caused by local transition, naturally we will think that whether the test cases can be calculated from the local transition structures of component automata.

First, let us see some definitions defined on the local transition structures of component automata. A local transition can only be activated at special local state and it can only change the state of its corresponding component. As shown in Fig. 2, the states are linked by transitions in the state transforming graph, so we get the following definition:

Definition 4. A State/Transition Path (S/T-Path) is a sequence of state transforming linked by transitions.

Since S/T-Paths are composed of linked state-transition pairs in component automata, they can interleave and branch off from other S/T-Paths. The construction of S/T-Path reflects the event-driven nature of the automata net model for networked software. Execution of networked software begins with an event, which we refer to as an input event.

An input event may trigger the execution of certain S/T-Path. The preorder S/T-Path may trigger the following S/T-Paths.

Based on the composition of S/T-path, the S/T-Path is divided into two classes.

Definition 5 Atomic S/T-Path. Atomic S/T-Path is defined as a formal state-transition pair which is a three-tuple

$$S/T_a = (A_i, q_i, t_i)\ . \tag{2}$$

1) A_i is the identification of component automaton which this atomic S/T-Path belongs to;
2) q_i is the state which this path can be enabled at;
3) t_i is the transition which will trigger this path.

In Fig. 2, $(A_1, q_0, a/c)$ is one atomic S/T-Path of A_1.

Following operators are defined for atomic S/T-Path in order to construct complex S/T-Path from atomic S/T-Path:

1) • stands for the sequential relationship between two atomic S/T-Paths. It is used to describe sequential control structure of the S/T-paths. For example, $P_1 \bullet P_2$ means that the execution of P_1 is ahead of the execution of P_2.
2) | stands for the selective relationship between two atomic S/T-Paths. It is used to describe optional control structure of the S/T-paths. For example, $P_1 | P_2$ means that either P_1 or P_2 will be carried out but not all.

3) $||$ stands for the parallel relationship between two atomic S/T-Paths. It is used to describe parallel control structure of the S/T-paths. For example, $P_1 || P_2$ means that both P_1 and P_2 will be carried out at same time.
4) $*$ stands for executing atomic S/T-Path repeatedly. For example, $P_1 * 2$ means P_1 will be carried out twice.
5) $\odot$ stands for the synchronous operation of two atomic S/T-Paths between component automata. For example, $P_1 \odot P_2$ means the output message of P_1 is the input message of P_2, and the transition of P_2 is triggered by this input message. P_1 belongs to A_i, and P_2 belongs to A_j, $i \neq j$.

Definition 6 Local S/T-Path. A reasonable composition of atomic S/T-Paths belonging to the same component automaton is defined as a local S/T-Path of software. An atomic S/T-Path is a special local S/T-Path.

In Fig. 2, $(A_1, q_0^1, a/c)$ is also a local S/T-Path of A_1 that is composed of a single atomic S/T-Path. $(A_2, q_0^2, c/d) \bullet (A_2, q_1^2, a/f)$ is a local S/T-Path that is composed of two atomic S/T-Paths.

Definition 7 Global S/T-Path. A reasonable composition of local S/T-Paths in the different component automata is defined as a global S/T-Path of networked software. In Fig. 2, $(A_1, q_0^1, a/c) || (A_2, q_0^2, c/d) \bullet (A_2, q_1^2, b/A_1!e) \odot (A_1, q_1^1, A_2?e/d)$ is a global S/T-Path.

The global S/T paths through combining atomic S/T-Paths with different operators are the test cases that can be used to test networked software.

3.2 Generation of Test Case

As discussed in previous section, we check the outputs of transitions when the inputs are applied. Test coverage rule is the foundation of test ending condition. In order to verify the correctness of the networked software, all output of local transitions should to be checked at least once so that all local transitions should be contained in certain transition sequences once. We only need to generate the transition sequence set that can coverage all local transitions. This transition sequence set is the test case set and each transition sequence is a test case which is formally defined as a global S/T-Path.

In this paper, we therefore proposed an S/T-Path coverage rule: test case set should make every atomic S/T-Path be covered once.

An atomic S/T-Path describes the state transition of certain software component when one transition is triggered at a certain local sate. The local transition is an ordered atomic S/T-Path sequence which is resulted in by the execution of component. The global S/T-Path is an ordered local S/T-Path sequence which is resulted in by dynamic interaction process among software components. So this rule not only covers the reaction of each component when certain input is applied but also the message sequence required in interaction process. In this way, the test case set based on this rule can not only check the local transition error but also the interaction error among different components.

In order to achieve this coverage rule, we need to define all the atomic S/T-Paths for networked software, and find a way of sequencing them. At same time, the minimum test cost is required. The problem is:

$$\begin{array}{l} Ap = \{atomatic\ S/T - paths\}; Tc = \{test\ cases\}; \\ \{paths(Tc)\} = Ap \qquad\qquad\qquad\qquad\qquad , \\ \min \sum_{i=1}^{n} C(Tc_i) \end{array} \tag{3}$$

where $C(Tc_i)$ means the test cost of the test case Tc_i. An exhaustive search is infeasible, so heuristics (such as greedy algorithm, Genetic algorithm and so on) may be applied.

4 Method Analysis

In this section, we will use our method to generate the test cases for the software shown in Fig. 2.

4.1 Test Case Set

Using our method, we should first define all the atomic S/T paths, and they are shown in Table 1.

Table 1. Atomic S/T paths

Component	
A_1	$(A_1, q_0^1, a/c); (A_1, q_1^1, A_2?e/d); (A_1, q_2^1, c/e); (A_1, q_3^1, g/h); (A_1, q_2^1, A_3?x/y)$
A_2	$(A_2, q_0^2, c/d); (A_2, q_0^2, a/c); (A_2, q_1^2, b/A_1!e); (A_2, q_1^2, a/f);$
	$(A_2, q_2^2, A_3?h/c); (A_2, q_3^2, h/A_4!d); (A_2, q_4^2, g/k)$
A_3	$(A_3, q_0^3, e/A_1!x); (A_3, q_1^3, e/c); (A_3, q_1^3, f/e); (A_2, q_4^3, a/b);$
	$(A_2, q_2^3, c/A_4!f); (A_3, q_3^3, a/b); (A_3, q_3^3, g/A_2!h)$
A_4	$(A_4, q_0^4, A_3?f/b); (A_4, q_0^4, e/x); (A_4, q_1^4, e/x); (A_4, q_2^4, A_2?d/f); (A_4, q_3^4, e/g)$

Secondly, in order to fulfill the S/T-Path coverage rule, we use equation (3) and genetic algorithm to generate the test case set. The generated test case set which has the least test cost is shown in Table 2. If this test case set is applied, all atomic S/T-Paths can be executed once at least.

4.2 State Number

From the example shown in Fig. 2, we can see that there are 18 states in the automata net model of the software. If we compute the compound automaton, the states set is computed by $Q = Q_1 \times Q_2 \times Q_3 \times Q_4$. The max state number of Q is

$|Q| = |Q_1| \times |Q_2| \times |Q_3| \times |Q_4| = 4 \times 5 \times 5 \times 4 = 400$.

Despite there may be some unreachable states, using the state simplifying algorithm the state apace can be reduced. But it is still more complex than the automata net model. At the same time, generating test cases from the local transition structure, we will not need to compute the compound automaton and simplify the unreachable and temporary states.

Table 2. Test case set

Test case	
1	$(A_1,q_0^1,a/c)\Vert(A_2,q_0^2,c/d)\bullet(A_2,q_1^2,b/A_1!e)\odot(A_1,q_1^1,A_2?e/d)\bullet$ $(A_3,q_0^3,e/A_1!x))\odot((A_1,q_2^1,A_3?x/y)\bullet(A_3,q_1^3,e/c)\bullet$ $(A_3,q_2^3,c/A_4!f)\odot(A_4,q_0^4,A_3?f/b)\bullet(A_4,q_1^4,e/x)\Vert(A_3,q_3^3,a/b)$
2	$(A_1,q_0^1,a/c)\Vert(A_2,q_0^2,c/d)\bullet(A_2,q_1^2,b/A_1!e)\odot(A_1,q_1^1,A_2?e/d)\bullet$ $(A_1,q_2^1,c/e)\bullet(A_1,q_3^1,g/h)\bullet(A_3,q_0^3,e/A_1!x))\odot(A_1,q_2^1,A_3?x/y)\bullet$ $(A_2,q_0^2,a/c)\Vert(A_3,q_1^3,f/e)\Vert(A_4,q_0^4,e/x)\bullet(A_3,q_3^3,g/A_2!h)\odot(A_2,q_2^2,A_3?h/c)\bullet$ $(A_2,q_3^2,h/A_4!d)\odot(A_4,q_2^4,A_2?d/f)\bullet(A_2,q_4^2,g/k)\Vert(A_4,q_3^4,e/g)$
3	$(A_2,q_0^2,c/d)\bullet(A_2,q_1^2,a/f)$

4.3 Unexecutable Test Sequence

Using our test case construction approach through combing atomic S/T-Paths, the test case is generated according to the local transition structures of software component and the interaction procedure between components, such as:

$$\begin{array}{l}(A_1,q_0^1,a/c)\parallel(A_2,q_0^2,c/d)\bullet(A_2,q_1^2,b/A_1!e)\odot(A_1,q_1^1,A_2?e/d)\bullet\\(A_3,q_0^3,e/A_1!x))\odot((A_1,q_2^1,A_3?x/y)\bullet(A_3,q_1^3,e/c)\bullet\\(A_3,q_2^3,c/A_4!f)\odot(A_4,q_0^4,A_3?f/b)\bullet(A_4,q_1^4,e/x)\parallel(A_3,q_3^3,a/b)\end{array}.$$

This is an executable transition sequence when the inputs are applied at different ports of software components.

Transition sequence

$$(A_1,q_0^1,a/c)\bullet(A_1,q_1^1,A_2?e/d)\bullet(A_2,q_0^2,c/d)\bullet(A_2,q_1^2,b/A_1!e)$$

cannot be executed, because the $(A_1,q_1^1,A_2?e/d)$ can be activated only when $(A_2,q_1^2,b/A_1!e)$ is executed. The unexecutable test cases are generated because the specification of interaction procedure between component automata is ignored when test generating.

5 Conclusion

When we use traditional test methods based on compound automata for networked software integration testing, we will suffer from the state combinatorial explosion problem. At same time, some generated test sequences may be unexecutable. In this paper, we proposed a new formal test case construction method based on automata net, in which we do not need to combine the automata models of software components. A formal definition S/T-Path was defined to describe the execution of software. The test cases are constructed through combining atomic S/T-Paths. These test cases are generated based on the local transition structures of software components and the interaction specification between components, so the state combinatorial explosion problem will not be encountered. The interaction procedure between components is considered when combing the local S/T-Paths, so all the test cases are executable.

The test cases generation and the benefits of our construction method in section 4 show that the test case construction through combing atomic S/T-Paths has better properties than the test case construction through searching the global transition structure of compound automaton. Thus, it is a promising way for networked software integration testing.

Acknowledgement

This work was supported in part by the National Key Technology R&D Program under Grant 2009BAG12A08, the R&D Foundation of the Ministry of Railways under Grant 2009X003 and the Research Foundation of Beijing National Railway Research and Design Institute of Signal and Communication.

References

1. Ghosh, S., Mathur, A.P.: Issues in testing distributed Component-Based Systems. In: Proceedings of the First International ICSE Workshop Testing Distributed Component-based System (1999)
2. Cho, I.-H., McGregor, J.D.: Component specification and testing interoperation of components. In: Proc. of the IASTED Int'l Conf., Software Engineering and Applications, pp. 27–31 (1999)
3. Cho, I.-H., McGregor, J.D.: A formal approach to specifying and testing the interoperation between components. In: Proceedings of the 38th annual on Southeast regional conference, pp. 161–170 (2000)
4. Luo, G., von Bochmann, G., Petrenko, A.: Test selection based on communicating nondeterministic finite-state machines using a generalized Wp-method. IEEE Transactions on Software Engineering 20(2), 149–162 (1994)
5. Hong, H.S., Lee, I., Sokolsky, O.: Automatic Test Generation from Statecharts Using Model Checking. In: Proceedings of FATES 2001, Workshop on Formal Approaches to Testing of Software, pp. 15–30 (2001)
6. Chow, T.S.: Testing Software Design Modeled by Finite Machines. IEEE Transaction on Software Engineering 4, 178–187 (1978)
7. Drusinsky, D.: Model checking of statecharts using automatic white box test generation. In: 48th Midwest Symposium on Circuits and Systems, pp. 327–332 (2005)
8. Biehl, M., Klarlund, N., Rauhe, T.: Algorithms for guided tree automata. In: Raymond, D.R., Yu, S., Wood, D. (eds.) WIA 1996. LNCS, vol. 1260, pp. 6–25. Springer, Heidelberg (1997)

9. Aho, A.V., Dahbura, A.T., Lee, D., Uyar, M.U.: An optimization technique for protocol conformance test generation based on UIO sequences and Rural Chinese Postman Tours. IEEE Transaction on Communication 39(11), 75–86 (1991)
10. Luo, G., von Bochmann, G., Petrenko, A.: Test selection based on communicating nondeterministic finite-state machines using a generalized Wp-method. IEEE Transactions on software Engineering 20(2), 149–162 (1994)
11. Hierons, R.M.: Checking States and Transitions of A Set of Communicating finite state machines. Microprocessors and Microsystems 24(9), 443–452 (2001)
12. Peeva, K.: Equivalence, reduction and minimization of finite automata over semirings. Theoretical Computer Science 88(2), 269–285 (1991)
13. Harju, T., Karhumäki, J.: The equivalence problem of multitape finite automata. Theoretical Computer Science 78(2), 347–355 (1991)
14. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to automata theory, languages, and computation, 2nd edn. Pearson Education, London (2000)
15. Fujiwara, S., von Bochmann, G., Khendek, F., et al.: Test selection based on finite state models. IEEE Transactions on Software Engineering 17(6), 591–603 (1991)
16. Choi, Y., Kim, D., Kim, J., et al.: Protocol test sequence generation using UIO and BUIO. In: 1995 IEEE International Conference on communications, vol. 1, pp. 362–366 (1995)

Automatic History Matching in Petroleum Reservoirs Using the TSVD Method

Elisa Portes dos Santos Amorim[1], Paulo Goldfeld[2], Flavio Dickstein[2],
Rodrigo Weber dos Santos[1], and Carolina Ribeiro Xavier[1]

[1] Dept. of Computer Science, Federal University of Juiz de Fora
Juiz de Fora, MG, Brazil
[2] Dept. of Applied Mathematics, Federal University of Rio de Janeiro
Rio de Janeiro, RJ, Brazil
elisaufjf@gmail.com, {flavio,goldfeld}@labma.ufrj.br,
rodrigo.weber@ufjf.edu.br

Abstract. History matching is an important inverse problem extensively used to estimate petrophysical properties of an oil reservoir by matching a numerical simulation to the reservoir's history of oil production. In this work, we present a method for the resolution of a history matching problem that aims to estimate the permeability field of a reservoir using the pressure and the flow rate observed in the wells. The reservoir simulation is based on a two-phase incompressible flow model. The method combines the truncated singular value decomposition (TSVD) and the Gauss-Newton algorithms. The number of parameters to estimate depends on how many gridblocks are used to discretize the reservoir. In general, this number is large and the inverse problem is ill-posed. The TSVD method regularizes the problem and decreases considerably the computational effort necessary to solve it. To compute the TSVD we used the Lanczos method combined with numerical implementations of the derivative and of the adjoint formulation of the problem.

Keywords: Reservoir simulation, History Matching, Optimization, TSVD, Adjoint formulation.

1 Introduction

Reservoir simulation is an essential tool extensively used by reservoir engineers. It is mostly employed to predict reservoir behavior under different circumstances, thus supporting decisions that frequently involve large financial costs. In order to use this tool properly different petrophysical properties of the reservoir must be well known, such as permeability and porosity. Unfortunately, direct measures of these properties are viable only near the wells. A way of estimating these properties is through the so called history matching process.

History matching process consists on the inverse problem of estimating reservoir properties through matching simulated data to reservoir history, which are available in reservoirs that are operating for some time. In this work we present

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 475–487, 2010.

a study for the automatic history matching based in a two-phase (oil/water), two dimensional reservoir model. The rate of oil production and the pressure measured at the wells are taken as the history of the reservoir. In this work, we aim to estimate the permeability distribution of the reservoir.

History matching may be seen as an optimization problem based on minimizing an objective function that measures the mismatch between reservoir history and simulated data. Efficient optimization methods are based on derivatives of the objective function. In this case, the derivative of data related to the parameters we aim to estimate. For the problem presented in this work, the calculation of this derivative related to the property of every gridblock that numerically represents the reservoir, is not computationally possible. We may find in the literature some ways to reduce the number of parameters to be estimated, as [1]. The approach presented in this work is the so called *truncated singular value decomposition* (TSVD), which reduces the search space of the minimization problem, reducing computational costs associated with its resolution. TSVD schemes used in history matching problems were discussed in [1], [2] and [3].

The paper is organized as follows: Section 2 introduces the direct problem formulation and implementation. Section 3 introduces the inverse problem theory and the TSVD method. Section 4 presents the derivative and adjoint formulations. Section 5 presents the methods and the computer platform used for the tests. Section 5 and 6 present the results and conclusion of this work, respectively.

2 Forward Problem

2.1 Theory

The problem treated in this paper is a two dimensional two-phase (water/oil) incompressible and immiscible porous media flow in a gravity-free environment. The system of partial differential equations which governs this flow is derived from the *law of mass conservation* and the *Darcy Law*.

The law of mass conservation for both phases is written as

$$\phi \partial_t (\rho_\alpha s_\alpha) + \nabla . (\rho_\alpha v_\alpha) = Q_\alpha \, . \tag{1}$$

where $\alpha = w$ denotes the water phase, $\alpha = o$ denotes the oil phase, ϕ is the porosity of the porous medium, and ρ_α, s_α, v_α and Q_α are, respectively, the density, saturation, volumetric velocity and flow rate in wells of the α-phase.

The volumetric velocity (v_α) is given by the Darcy law as follows

$$v_\alpha = \frac{K k_{r\alpha}(s_\alpha)}{\mu_\alpha} \nabla p_\alpha \, . \tag{2}$$

where K is the effective permeability of the porous medium, $k_{r\alpha}$ is the relative permeability of α-phase, which is a function that depends on saturation, and μ_α and p_α are, respectively, viscosity and pressure of the α-phase. In this work we consider that the capillary pressure is null, that is, $p_w = p_o$. So, from now on we will refer to pressure simply as p.

In addition to Equations 1 and 2 we have

$$s_w + s_o = 1 \; . \tag{3}$$

We introduce the phase mobility and transmissibility functions, respectively

$$\lambda_\alpha(s) = \frac{k_{r\alpha}(s)}{\mu_\alpha} \; . \tag{4}$$

$$T_\alpha(s) = K\lambda_\alpha \; . \tag{5}$$

where $s = s_w$ from now on. The volumetric velocity can then be written as $v_\alpha = -T_\alpha \nabla p$.

We assume that the phases density and viscosity are constant and get

$$\begin{cases} \phi\rho_w\partial_t s_w + \rho_w \nabla v_w = Q_w \\ \phi\rho_o\partial_t s_o + \rho_o \nabla v_o = Q_o \; . \end{cases} \tag{6}$$

Now we can divide the equations in 6 by ρ_α and sum both. Using Eq. 3 we arrive at the following system

$$\begin{cases} \phi\partial_t s + \nabla v_w = q_w \\ \nabla v_t = q_t \; . \end{cases} \tag{7}$$

where $q_\alpha = \frac{Q_\alpha}{\rho_\alpha}$ is the flow rate density of α-phase, $q_t = q_w + q_o$ and $v_t = v_w + v_o$.

Defining total mobility as $\lambda_t = \lambda_w + \lambda_o$ we introduce the fractional flow functions as

$$f(s) = \frac{T_w}{T_t} = \frac{\lambda_w}{\lambda_t} \tag{8}$$

System 6 is then rewritten as

$$\begin{cases} \phi\partial_t s - \nabla(f(s)T_t(s)\nabla p) = q_w \\ -\nabla(T_t(s)\nabla p) = q_t \; . \end{cases} \tag{9}$$

To complete the model the boundary conditions must be specified. In this paper we consider no flow boundary condition, which means that

$$v_\alpha.\nu = 0, x \in \partial\Omega \tag{10}$$

where ν is the outer unit normal to the boundary $\partial\Omega$ of the domain Ω. Finally we define the initial condition given by

$$s(x,0) = s_0(x), x \in \Omega \tag{11}$$

The forwad problem treated on this paper is the system of partial differential equations given by 9 with boundary and initial condition given by 10 and 11 respectively.

There are several approaches for solving the described system [4]. In this work we used the IMPES method, which can be found in details in [5].

3 Inverse Problem

In this work, the inverse problem proposed aims to estimate the absolute permeability field of a reservoir by history-matching its observed data. The observed data here is the oil flow rate in production wells and the pressure in both, production and injection wells. In the discretized forward problem stated in Section 2, we assume that permeability is constant on each grid block.

Parameter estimation is essentially an optimization problem whereby the unknowns parameters are obtained by minimizing an objective function [8]. We denote by K the vector of permeability to be determined and by O the vector of observations and define as $u = (s, p)$ the vector of the forward problem unknowns (saturation and pressure). We have that u depends on the permeability $u = u(K)$ and O depends on both, permeability and u. Thus,

$$O(K) = O(K, u(K)).$$

If $\bar{O}$ is the vector with the real observations we can determine K using the least square formulation

$$f(K) = \|O(K) - \overline{O}\|^2, \tag{12}$$

and

$$min_K f(K). \tag{13}$$

Note however that this is a constrained minimization problem since permeability is a strictly positive property. In this work, we transformed this problem in an unconstrained minimization problem via the change of variable $m_i = ln(K_i)$. From now on, the parameters to be estimated are those of the vector m. There are several ways for solving this optimization problem. In this work we present the Gauss-Newton combined with the TSVD method for solving this problem.

3.1 The TSVD Method

As presented in Section 3, our problem consists on the determination of a permeability field in a way that the model described by System 9 reproduces the observed data. Let n_o and n_m be the number of observations and the number of parameters to be estimated, respectively. Let the residue $r_i(m) = \bar{O}_i - O_i(m), i = 1, ..., n_o$ be the model prediction error associated with ith observation.

The problem of unconstrained non-linear optimization consists on finding a global minimum for the sum of squares of n_o non-linear functions

$$\min_{m \in \mathbb{R}^{n_m}} f(m), \qquad f(m) = \frac{1}{2}\|r_i(m)\|_2^2 = \frac{1}{2}\sum_{i=1}^{n_o} r_i^2(m). \tag{14}$$

Equation 14 may be rewritten by approximating r_i by $r_i(m_c) + J(m_c)(m - m_c)$, where $J(m_c) = -O'(m_c)$ is the Jacobian of r and m_c is the current estimative of m. Using this approximation it is possible to solve the nonlinear least square problem by a sequence of linear least square problems of the form

$$\min_{m} \|r(m_c) - O'(m_c)(m - m_c)\|_2, \tag{15}$$

This approach is called the Gauss-Newton method.

A classical method for solving the linear least square problem (15) is the singular value decomposition (SVD) method. The SVD of a matrix A provides a diagonal form of A under an orthogonal equivalence transformation [9]. Consider a matrix $A \in \mathbb{R}^{m\times n}$ of rank r. Then the SVD of A is a decomposition of the form

$$A = U\Sigma V^T = \sum_{i=1}^{n} u_i \sigma_i v_i^T, \tag{16}$$

where $U = (u_1, ..., u_n) \in \mathbb{R}^{m\times n}$ and $V = (v_1, ..., v_n) \in \mathbb{R}^{n\times n}$ are the left and right singular vectors of A, respectively and are matrices of orthonormal columns. The diagonal matrix $\Sigma = diag(\sigma_1, ..., \sigma_r)$ has nonnegative diagonal elements appearing in nonincreasing order such that $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r$. The numbers σ_i are called the singular values of A and the vectors u_i and v_i are the left and right singular vectors of A, respectively [10].

Thus the *pseudoinverse* of A is given by

$$A^+ = \sum_{i=1}^{r} v_i \sigma_i^{-1} u_i^T \tag{17}$$

and the least square solution x_{LS} to the least square problem $min\|Ax - b\|_2$ is given by

$$x_{LS} = A^+ b = \sum_{i=1}^{r} \frac{u_i^T b}{\sigma_i} v_i. \tag{18}$$

Divisions by small singular values may amplify the high-frequency components in b [10]. Problems presenting such small singular values are called *numerically rank-deficient*. They may be regularized by considering the given matrix A as a noisy representation of a mathematically rank-deficient matrix, and replace A by a matrix that is close to A and mathematically rank-deficient. The standard choice is the $rank - k$ matrix A_k defined as

$$A_k \equiv \sum_{i=1}^{k} u_i \sigma_i v_i^T, \tag{19}$$

that is, we replace the small nonzero singular values $\sigma_{k+1}, ..., \sigma_r$ with exact zeros.

When A is replaced by A_k we obtain a new least square problem $\min \|A_k x - b\|_2$. The minimum-norm solution x_k to this problem is given by

$$x_k = A_k^+ b = \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i} v_i. \tag{20}$$

The solution x_k is referred to as the *truncated* SVD solution. The complete method is called *truncated* SVD (TSVD) and the matrix A_k is called the TSVD matrix [10].

To compute the TSVD matrix we use the Lanczos Method [7], which requires the calculation of the A and its adjoint A^T applied to a vector. In this work $A = O'(m)$.

3.2 The Derivative and the Adjoint Formulations

As seen in Section 3.1, in order to use the TSVD method we need to have two important tools: the derivative and the adjoint of the PDE system (9) applied to a vector, that is, $O'(m)z$ and $(O'(m))^T w$.

The Derivative. The vector of observations O may be written as $O(m, u(m))$, where $u = (s, p)$ is the vector of unknowns of the System (9). The derivative of O applied to a vector z is given by

$$O'(m)z = (\partial_m O(m, u(m)) + \partial_u O(m, u(m))u'(m))z. \tag{21}$$

The term $u'(m)$ is the most complex of the above formulation, because $u(m)$ is defined implicitly by the system of PDEs (9). $u'(m)$ involves the calculation of the Jacobian J of the system (9) related to u applied to a vector. Using Taylor series, we find that the resolution of $J(ds, dp)$ consists in the resolution of a system of PDE, given by

$$\begin{cases} \phi\frac{\partial ds}{\partial t} = \nabla.(f(\bar{s})T_t(\bar{s})\nabla dp) + \nabla((f(\bar{s})T_t(\bar{s}))' ds\nabla\bar{p}) + f'(\bar{s})dsq_t + y_s \\ -\nabla((T_t'(\bar{s})ds\nabla\bar{p}) - \nabla.(T_t(\bar{s})\nabla \bar{d}p) = y_p, \\ v_\alpha.\nu = 0, x \in \partial\Omega, \\ ds(x, 0) = 0, \end{cases} \tag{22}$$

which is a system equivalent to (9) in terms of computational complexity.

3.3 The Adjoint

Using (21) we may write $(O'(m)^T)w$ as

$$(O'(m)^T)w = (\partial_m O(m, u(m))^T - ((u'(m))^T \partial_u O(m, u(m))^T)w. \tag{23}$$

For the same reason pointed on the derivative case, the most complex term of this formulation is $u'(m)^T$ and it naturally involves the calculation of J^T applied to a vector. Using $\langle Jz, w\rangle = \langle z, J^T w\rangle$, we define $J^T(ws, wp)$ as

$$\begin{cases} -\phi\dfrac{\partial ws}{\partial t} + \nabla.(T_t(\bar{s})\nabla wp) + (f(\bar{s})T_t(\bar{s}))'\nabla\bar{p}\nabla ws \\ \qquad\qquad -T_t'(\bar{s})\nabla\bar{p}\nabla wp - f'(\bar{s})wsq_t = zs \\ -\nabla.(f(\bar{s})T_t(\bar{s})\nabla ws) + \nabla.(T_t(\bar{s})\nabla wp) = zp \\ v_\alpha.\nu = 0, x \in \partial\Omega \\ ws(x, T) = 0. \end{cases} \tag{24}$$

This system is also equivalent to (9). Thus, the resolution of the derivative and the adjoint demands an equivalent effort to solve the forward problem.

4 Methods

4.1 Implementation Details and Computer Platform

The numerical solution of the forward problem was implemented in C++. To solve the linear systems associated to the discretization of the Partial Differential Equations the PETSc library [6] was used.

The experiments were performed in a AMD Turion 64 X2 TL-60 2000 MHz processor with 2GB of RAM.

4.2 Numerical Experiments

The reservoir simulation we consider in this work is the classical five-spot configuration with 4 production wells (P1,P2,P3 and P4) in the corners of the reservoir and one injection well (I) in its center. The reservoir is a square of sizes equal to $200m$ and depth equal to $20m$. This configuration in illustrated by Figure 1.

The injection well injects a total of $400m^3$ per day, and all four production wells produce at a fixed rate of $100m^3$ per day. The reservoir's history is given by the oil production of the production wells and the pressure observed from the five wells during 400 days of simulation, by each 10 days. This means we have 40 days of observation, with 160 flow rate data and 200 pressure data, giving a total of 360 observation data.

The other parameters of the model are: porosity, constant equal to 0.2, relative permeability, given by the Corey curve; irreducible water saturation $s_{iw} = 0.2$ and residual oil saturation $s_{ro} = 0.2$. In this work, two different synthetic histories were generated from different permeability field configurations, one with 21×21 grid blocks and the other one with 51×51.

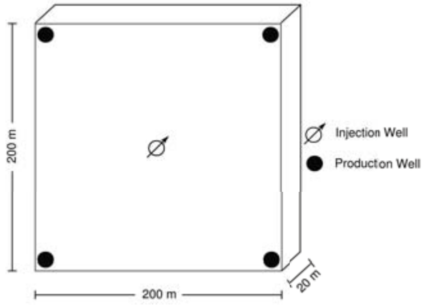


Fig. 1. The five-spot configuration

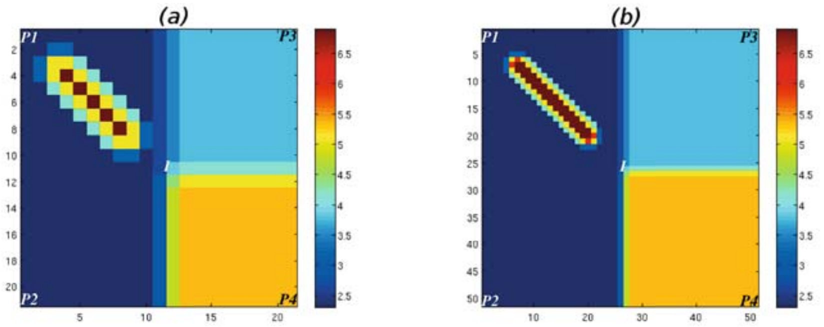


Fig. 2. Permeability fields used to generate the history. (a) is discretized with a gridblock of 21 × 21 and (b) is discretized with a gridblock of 51 × 51.

Both permeability fields used on each example present three well defined regions with different permeability values and a high permeability channel connecting the injection well to one of the production wells (P1), as may be seen in Figure 2. This may be an interesting configuration to analyze, since it presents some challanges for the optimization, such as reproducing the channel and identifying different permeability regions. The two synthetic histories obtained by simulating these permeability fields are the targets of two different history-matching problems.

Number of Singular Values. In order to evaluate the TSVD method for these problems, we used four different singular values configurations. In three of them it was employed a fixed number of singular values for each iteration. The numbers 7, 15 and 25 were chosen so we could analyze the impact of more and less singular values in the results. Additionally, it was employed an increasing number of singular values strategy. This strategy consists on fixating the number of singular values used and run the optimizer until it converges. When it happens, the optimal permeability field found during the optimization process is used as the initial guess for another optimization, using a greater number of singular values. The number of singular values used by this strategy was 2, 4, 8, 12, 16, 20 and 25.

Objective Function, Stop Criteria and Initial Guess. As we have two different types of observation (pressure and flow rate), the objective function includes a weight to put the observations in the same magnitude. The objective function is then written as

$$f(m) = \frac{1}{2}(\|\alpha_q(\bar{O}_q - O_q(m))\|^2 + \|\alpha_p(\bar{O}_p - O_p(m))\|^2),$$

where subscripts p and q are related to pressure and flow rate, respectively; α is the weight used and O is the observation. In this work we used $\alpha_\beta = \frac{1}{\|\bar{O}_\beta\|}$, where $\beta = p, q$.

Convergence is achieved when a candidate solution m produces $f(m) < 1.0e-6$, or $\|\nabla f(m)\| < 1.0e-6$ or the number of iterations is greater than 50.

A homogeneous permeability field equal to the geometric mean of the objective permeability was used as the initial guess for both experiments.

5 Results

In this section we present the results of the history matching for each case. Let $f_r = \frac{f_{final}}{f_{initial}}$ be the relative value of the objective function and $f_{r\alpha} = \frac{f_{initial\alpha}}{f_{final\alpha}}$ be the relative value of the objective function related to α, where $\alpha = q, p$. NI is the number of iterations in the optimizer and T(s)/I is the average time spent on each optimization iteration. Table 1 presents the results obtained by the optimization of the first permeability field.

Table 1. Permeability field (a)

SV	f_r	f_{rq}	f_{rp}	NI	T(s)/I
7	1.5306e-02	5.1321e-02	7.9203e-03	19	12.10
15	1.4238e-03	2.2789e-03	1.2485e-03	23	25.42
25	1.0164e-04	1.7100e-04	8.7411e-05	30	36.29
Increasing	7.6702e-04	1.6594e-04	6.0108e-04	74	23.55

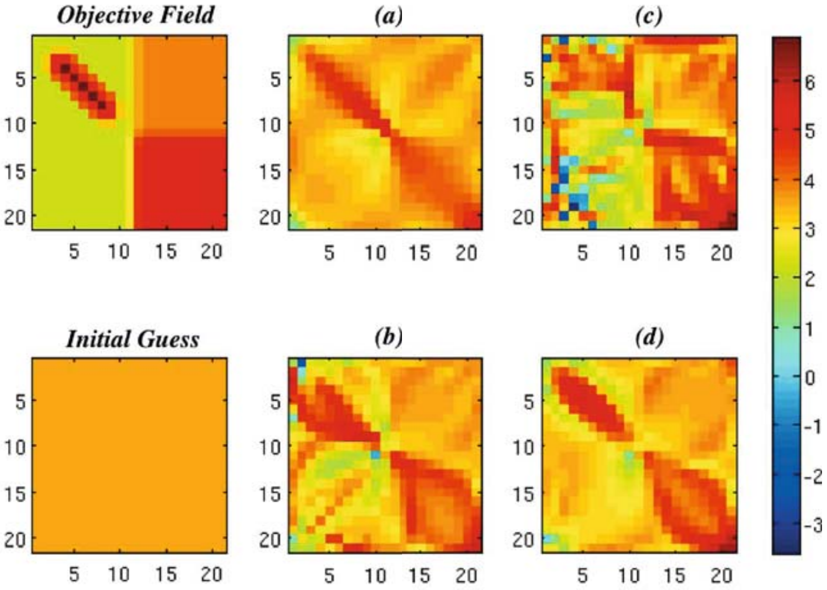


Fig. 3. Permeability fields for the first case. (a) 7 SV; (b) 15 SV; (c) 25 SV; (d) Increasing SV strategy.

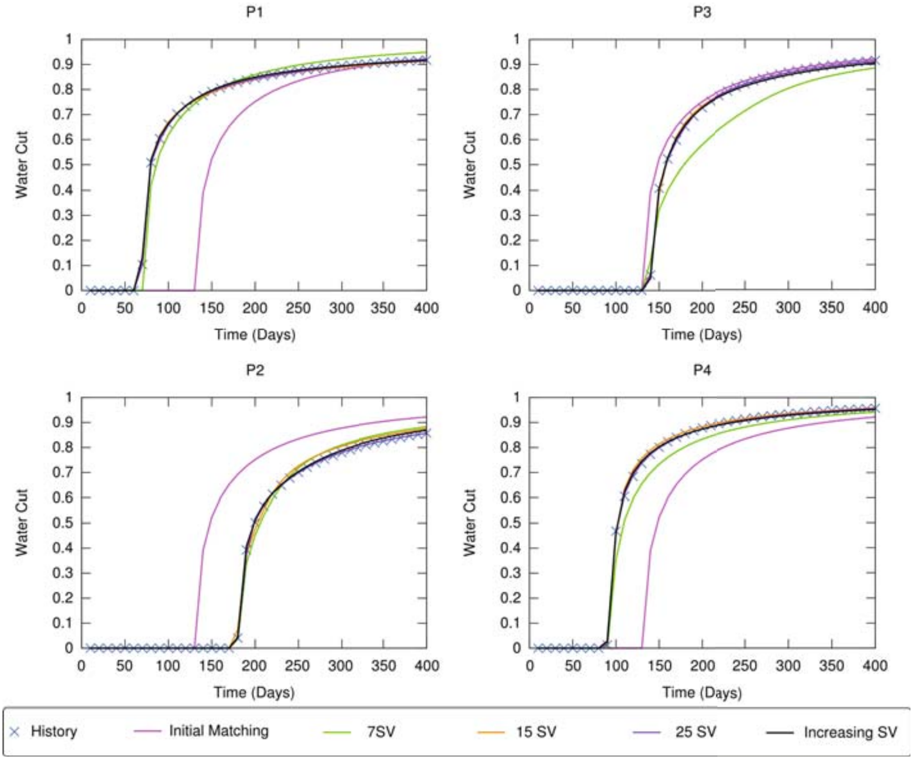


Fig. 4. Permeability fields for the first case. (a) 7 SV; (b) 15 SV; (c) 25 SV; (d) Increasing SV strategy.

Quantitatively, the best results were achieved in the case of 25 singular values and with the increasing singular values strategy, where the final objective function was decreased to 1.0164e-04 and 7.6702e-04 times the initial objective function, respectively. It is also interesting to note that with 25 singular values we got a more expensive optimizer iteration, as we could expect. But the increasing singular value strategy was the most expensive, if we consider the entire optimization process, as it required more optimizer iterations. Figure 3 presents the permeability fields found for each case.

Visually, every permeability field found represents the synthetic field somehow. The channel was better reproduced in the case with increasing singular values strategy, while the different blocks were better determined in the case with 7 singular values.

It is well known that there is no unique solution for the history matching problem, that is, an infinite number of equally good solutions exist [11]. Results on Figure 3 are quite distinct, specially the one produced with 25 singular values. However, Figure 4 presents the flow rate matching, represented by the water cut ($\frac{Q_w}{Q_t}$), for each singular value strategy and it is clear that what we got is an almost perfect matching, except for the 7 singular values strategy.

Table 2 presents the results obtained for the second permeability field of Figure 2.

Again, the best results were obtained with the increasing singular values strategy and the 25 fixed singular values. As we observed in the previous example, and as expected, the optimizer using 25 singular values was more expensive per iteration while the increasing value strategy was more expensive considering the entire optimization process. Figure 5 presents the permeability fields found for this second experiment.

Similarly to the case with a gridblock discretization of 21 × 21, we may observe that each singular value strategy was able to reproduce some important characteristics of the permeability field, specially the channel. Furthermore, the permeability fields in this case are pretty similar to those found on previous case (although the permeability fields used to generate the history are not strictly the same).

Table 2. Permeability field (b)

SV	f_r	f_{rq}	f_{rp}	NI	T(s)/I
7	4.6349e-02	2.4769e-01	9.9379e-03	20	284.65
15	9.3485e-03	4.5579e-02	2.7965e-03	16	481.26
25	1.0048e-03	1.6795e-03	8.8279e-04	36	833.15
Increasing	9.2601e-04	7.3668e-04	9.6024e-04	103	370.51

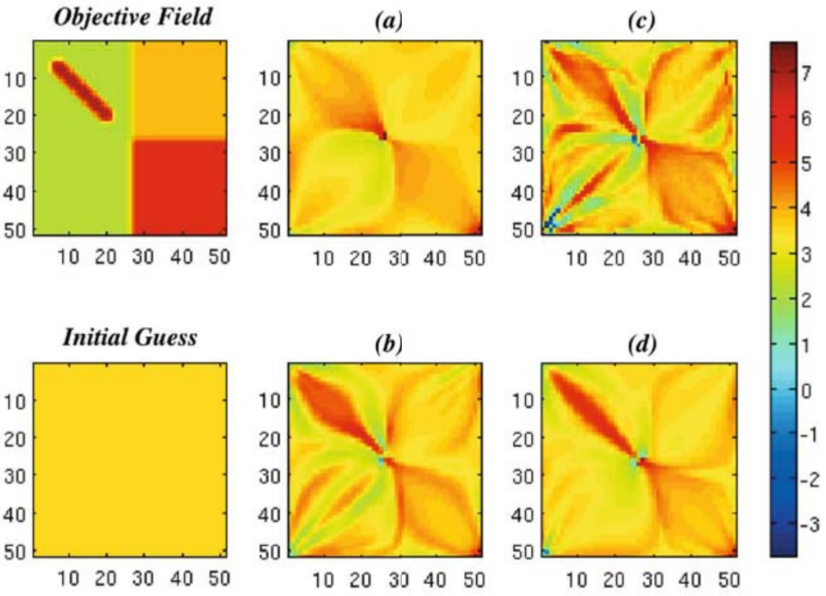


Fig. 5. Permeability fields for the second case. (a) 7 SV; (b) 15 SV; (c) 25 SV; (d) Increasing SV strategy.

Quantitatively we may observe by Tables 1 and 2 that the objective function in this example did not decreased by the same rate as with less parameters. However, in the enginering's point of view, the results are qualitatively very similar, since the reservoir was well characterized in both examples. In this light, these results are very important because they suggest that, no matter how fine the grid is, the number of singular values required to perform the history matching will not increase.

6 Conclusion

In this work we presented the TSVD method combined with the Gauss-Newton method to solve the history matching problem in the context of reservoir simulators. Our goal was to estimate permeability characteristics of a reservoir using the oil rate and pressure measured in wells as the history.

This work presented some experiments based on synthetic permeability fields. It was observed that in most cases the optimizer was able to reproduce important characteristics of the permeability field. Besides, we were able to verify through an example using more parameters that the required number of singular values does not depend on the number of parameters we aim to estimate.

In general, the developed tool was able to identify a field that reproduced well the history and some characteristics of the permeability field. This is a completely automatic tool, which is a very positive feature in a process that is traditionally very challenging.

Acknowledgement

The authors would like to thank PETROBRAS for the financial support.

References

1. Tavakoli, R., Reynolds, A.C.: History Matching With Parameterization Based on the SVD of a Dimensionless Sensitivity Matrix. SPE J. SPE-118952-PA (2009)
2. Rodrigues, J.R.P.: Calculating derivatives for history matching in reservoir simulators. In: SPE Reservoir Simulation Symposium (2005)
3. Rodrigues, J.R.P.: Calculating derivatives for automatic history matching. Computational Geosciences 10, 119–136 (2006)
4. Chen, Z., Huan, G., Ma, Y.: Computational Methods for Multiphase Flows in Porous Media. Society for Industrial and Applied Mathematics, Philadelphia (2006)
5. Chen, Z., Huan, G., Li, B.: An improved IMPES method for two-phase flow in porous media. Transport in Porous Media 32, 361–376 (2004)
6. Balay, S., Buschelman, K., Gropp, W.D., Kaushik, D., Knepley, M., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory (2002)
7. Komzsik, L.: The Lanczos Method, Evolution and Application. Society for Industrial Mathematics, Philadelphia (1987)

8. Englezos, P., Kalogerakis, N.: Applied Parameter Estimation for Chemical Engineers. Marcel Dekker, New York (2001)
9. Bjork, A.: Numerical Methods for Least Square Problems. Society for Industrial and Applied Mathematics, Philadelphia (1996)
10. Hansen, P.C.: Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion. Society for Industrial and Applied Mathematics, Philadelphia (1998)
11. Oliver, D.S., Reynolds, A.C., Liu, N.: Inverse Theory for petroleum reservoir characterization and history matching. Cambridge University Press, Cambridge (2008)

An Efficient Hardware Architecture from C Program with Memory Access to Hardware

Akira Yamawaki[1], Seiichi Serikawa[1], and Masahiko Iwane[2]

[1] Kyushu Institute of Technology
1-1, Sensui, Tobata, Kitakyushu, 804-8550, Japan
yama@ecs.kyutech.ac.jp
[2] Yuundo, Ltd., Japan

Abstract. To improve the performance and power-consumption of the system-on-chip (SoC), the software processes are often converted to the hardware. However, to extract the performance of the hardware as much as possible, the memory access must be improved. In addition, the development period of the hardware has to be reduced because the life-cycle of SoC is commonly short. This paper proposes a design-level hardware architecture (semi-programmable hardware: SPHW) which is inserted onto the pass from C to hardware. On the SPHW, the memory accesses and buffers are realized by the software programming and parameters respectively. By using the SPHW you can easily develop the data processing hardware containing the efficient memory access controller at C-level abstraction. Compared with the conventional cases, the SPHW can reduce the development time significantly. The experimental result also shows that you can employ the SPHW as the final product if the memory access latency is hidden enough.

1 Introduction

Recently, the functionality of the system-on-chip (SoC) is rapidly growing. The SoC has to quickly handle heavy applications like the movie, the high-resolution image and the audio with the low-power consumption. To answer such requests, the SoC designer often converts the software processing to the hardware. However, in order to extract the performance of the hardware as much as possible, not only the data processing but also the memory access has to be improved. In addition, since the life-cycle of the SoC is commonly short, the development period of the hardware modules has to be reduced.

To reduce the development period of the hardware, the high-level synthesis (HLS) tools which convert the C program to the hardware have been researched and developed [1,3,4,5,6,8,9]. Although they have concentrated on the data processing well, improving memory access has not been paid much attention. For example, some compilers of the HLS support only the dedicated memory access pattern [5,8,9]. The memory access patterns vary according to the user, the application programs and the buffering methods. Thus, the memory accesses are hard to be treated systematically by an algorithmic way. In addition, the

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 488–502, 2010.

memory access latency cannot be hidden by the data-prefetching [10] implicitly [1,3,4,5,6,8,9]. To hide the memory latency, the hardware has to be written skillfully in the C description with the deep knowledge of the HLS and the target device. Then, it is desired that the HLS tool used will generate the hardware including an efficient memory access controller. In spite of having raised the design abstraction, the time and effort may be comparable to designing a memory access circuit from scratch in a hardware description language (HDL).

From the viewpoint of the C language concept, the variables and arrays used are put into the linear memory. Thus, we think that generating efficient memory access controller for the linear memory from the C description is very important for the performance and the concept. This paper proposes a design-level hardware architecture (semi-programmable hardware: SPHW) which is inserted onto the pass of the HLS from C to hardware. In the SPHW, the memory accesses are realized by the software programming and the buffer used by the data-prefetching is realized via parameters. By using the SPHW, you can easily develop the data processing hardware with the data-prefetching mechanism which hides the memory access latency by overlapping to the data processing at the high abstraction of the C level.

The rest of the paper is organized as follows. Section 2 shows the design flow with the SPHW and describes the overview of the SPHW. Section 3 maps some application programs which show the different memory accesses onto a prototype of the SPHW. Section 4 evaluates the effect of the SPHW to the development period, the hardware overhead and the performance. Finally, Section 5 concludes our paper.

2 Semi-programmable Hardware Overview

2.1 Design Flow

Fig. 1 shows the framework of the design flow which employs the SPHW. In the SPHW, the memory accesses are implemented by the software programming to the load/store unit (LSU). The reconfigurable register file (RRF) is configured by the parameters to implement the optimum buffer. The data processing unit (EXU) streamly processes the sequential data on the RRF. The memory data which shows the sophisticated access patterns are put into the RRF as the stream data by the LSU. An HLS tool has only to consider the stream data on the RRF; it is good at converting the stream processing to the hardware [1,3,4,5,6,8,9]. Since the LSU and the EXU are executing individually across the RRF, the memory access by the LSU can overlap onto the data process by the EXU. By using the SPHW, the designer can design the hardware with the data prefetching mechanism [10] easily in the high-level description using the program and parameters.

2.2 SPHW Organization

Fig. 2 shows the organization of the SPHW. The load/store unit (LSU) loads the memory data into the input data buffer register (DBRI). The LSU stores the

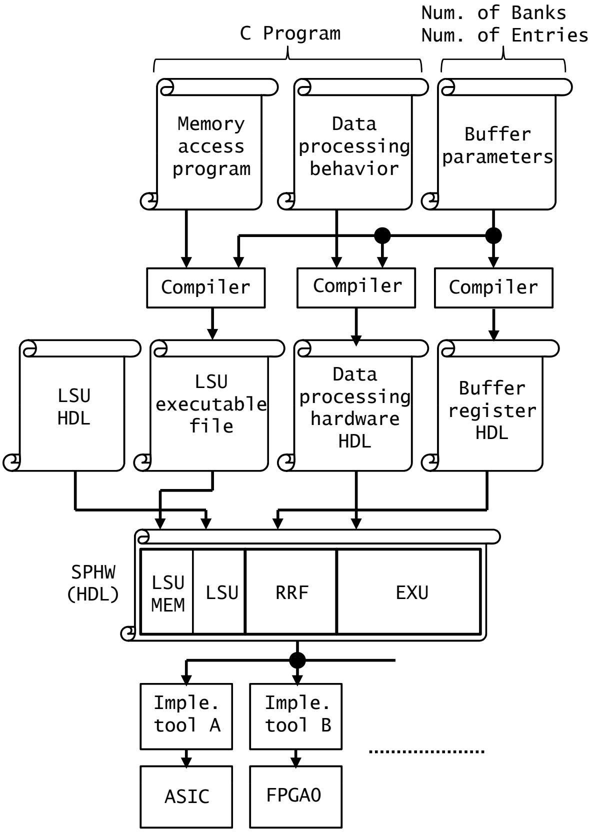


Fig. 1. Design Flow Framework with SPHW

processed data from the output data buffer register (DBRO) into the memory. The LSU executes the program in the LSU memory (LSUMEM).

The DBRI and the DBRO in the reconfigurable register file (RRF) are the register files that have one or more banks. Each of banks contains one or more entries. The number of the banks and the entries are configurable by the parameters. Thus, the designer can configure the RRF as the suitable buffer for the data processing hardware on the execution unit (EXU) by parameters. The mailbox (MB) is control/status registers for the SPHW. The external modules can check the statuses of the SPHW via mailboxes. The parameters required for the SPHW execution can be set via the mailboxes. To implement an interrupt, the mailboxes can be directly outputted to the outside of the SPHW. The general purpose register (GPR) is used by the LSU and the EXU.

The EXU is data processing hardware. Basically, the EXU has the finite state machine (FSM), the working registers (WR) and the data path. The FSM has the states ($\langle \mathrm{EXE_i} \rangle$) to control the data path. In addition, the states ($\langle \mathrm{SYNC_i} \rangle$) to synchronize the LSU are inserted.

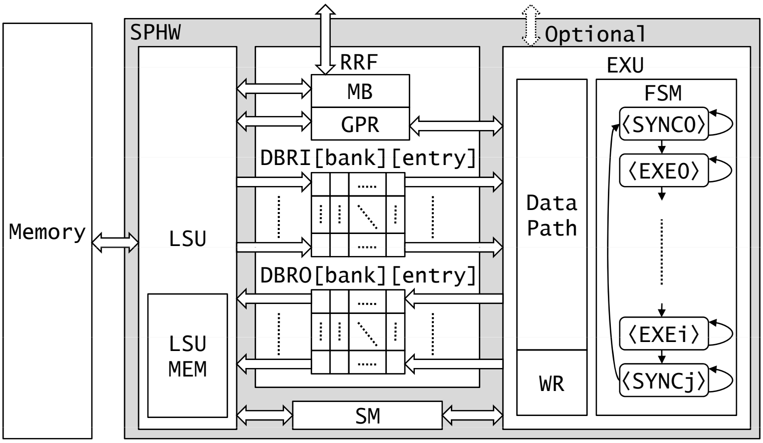


Fig. 2. SPHW Organization

The synchronization mechanism (SM) performs the producer-consumer synchronization between the LSU and the EXU. In the producer-consumer synchronization, the producer performs the release synchronization to invoke the consumer. The consumer performs the wait synchronization to wait until the producer issues the release synchronization. The LSU can perform the synchronization simultaneously with the load/store operations.

The reconfigurable register file and the synchronization mechanism provide the EXU with the memory and bus-tolerant simple interface for the data access.

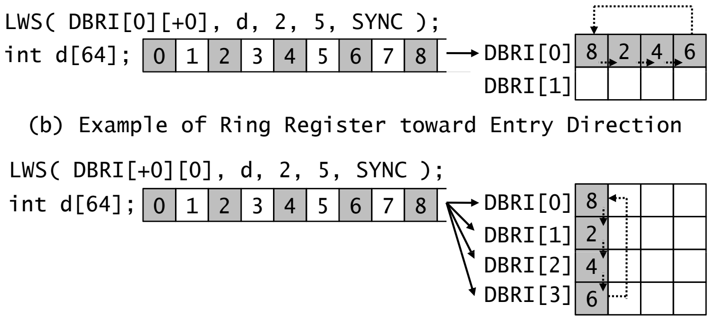
2.3 Memory Access

The LSU has the load/store instructions per the word and the line containing continuous words. Each of instructions can specify the number of transfers and the stride width. That is, the LSU can perform the gather/scatter operations by one instruction. Since the load/store instructions have the synchronization field, the synchronization can be also performed simultaneously with the memory access.

In Fig. 3 (a), the LWS and LLS are the load instructions. The SWS and SLS are the store instructions. The LWS and SWS transfer the words. The LLS and SLS transfer the lines with some continuous words. Regardless of the word transferring or the line transferring, the pointers to the bank and entry of the data buffer registers are incremented per the word transfer. Which pointer is incremented is specified by '+' in the brackets as shown in Fig. 3 (a). When '+' is described at the left bracket, the bank pointer is incremented. When '+' is described at the right bracket, the entry pointer is incremented. The Rm, the Rs and the Rn are the general purpose register or the mailbox. They hold the memory address, the stride width and the number of transfers respectively. The stride width in the Rs is added to the Rm that holds the memory address.

```
                                   bank  entry
Load Line Stream  : LLS(DBRI[+i][j ], Rm, Rs, Rn, SYNC);
                    LLS(DBRI[i ][+j], Rm, Rs, Rn, SYNC);
Load Word Stream  : LWS(DBRI[+i][j ], Rm, Rs, Rn, SYNC);
                    LWS(DBRI[i ][+j], Rm, Rs, Rn, SYNC);
Store Line Stream : SLS(DBRO[+i][j ], Rm, Rs, Rn, SYNC);
                    SLS(DBRO[i ][+j], Rm, Rs, Rn, SYNC);
Store Word Stream : SWS(DBRO[+i][j ], Rm, Rs, Rn, SYNC);
                    SWS(DBRO[i ][+j], Rm, Rs, Rn, SYNC);
```

(a) Load/Store Operations



```
LWS( DBRI[0][+0], d, 2, 5, SYNC );
int d[64];
```

(b) Example of Ring Register toward Entry Direction

```
LWS( DBRI[+0][0], d, 2, 5, SYNC );
int d[64];
```

(c) Example of Ring Register toward Bank Direction

Fig. 3. LSU Load/Store Operations

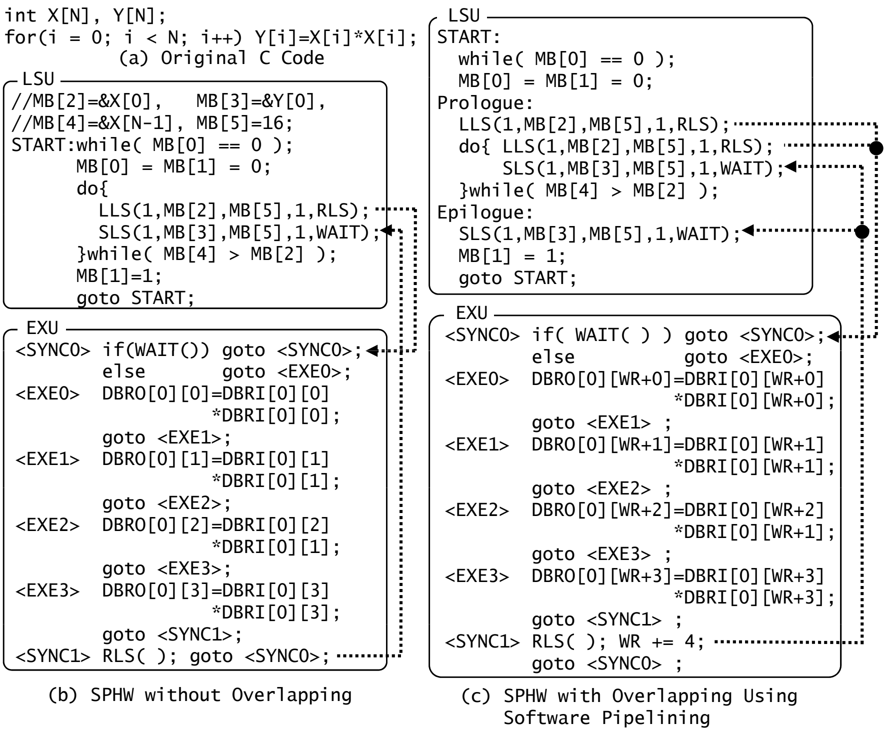
The LWS and SWS add the Rs to the Rm per the word transferring. The LLS and SLS add the Rs to the Rm per the line transferring. The synchronization between the LSU and the EXU is specified by the synchronization field (SYNC).

Fig. 3 (b) shows an example that the LSU gathers the distributed words on the memory into the input data buffer register (DBRI) as the continuous words. We assume that the width of the DBRI is 4byte, the number of banks of the DBRI is 2 and each bank contains 4 entries. The dotted arc means the order of which the array elements are loaded into the DBRI. The LWS loads the array every other element into the DBRI. By incrementing the entry pointer, the ring register toward the entry direction is implemented.

Fig. 3 (c) shows an example that the LSU loads the array elements into the DBRI toward the bank direction. In this example, the DBRI has 4 banks containing 4 entries. As Similar to the Fig. 3 (b), the LSW can implement the ring register toward the bank direction.

2.4 Simple Example

Let us show an overview of mapping the program to the SPHW using the simple example shown in Fig. 4 (a). Fig. 4 (b) shows the straight-forward mapping. Fig. 4 (c) shows an extended version of Fig. 4 (b) to hide memory access latency. The pseudo code based on the C language is used to describe the LSU program and the EXU behavior.



```
int X[N], Y[N];
for(i = 0; i < N; i++) Y[i]=X[i]*X[i];
```

(a) Original C Code

```
LSU
//MB[2]=&X[0],   MB[3]=&Y[0],
//MB[4]=&X[N-1], MB[5]=16;
START:while( MB[0] == 0 );
      MB[0] = MB[1] = 0;
      do{
        LLS(1,MB[2],MB[5],1,RLS);
        SLS(1,MB[3],MB[5],1,WAIT);
      }while( MB[4] > MB[2] );
      MB[1]=1;
      goto START;
```

```
EXU
<SYNC0> if(WAIT()) goto <SYNC0>;
        else       goto <EXE0>;
<EXE0>  DBRO[0][0]=DBRI[0][0]
                  *DBRI[0][0];
        goto <EXE1>;
<EXE1>  DBRO[0][1]=DBRI[0][1]
                  *DBRI[0][1];
        goto <EXE2>;
<EXE2>  DBRO[0][2]=DBRI[0][2]
                  *DBRI[0][1];
        goto <EXE3>;
<EXE3>  DBRO[0][3]=DBRI[0][3]
                  *DBRI[0][3];
        goto <SYNC1>;
<SYNC1> RLS( ); goto <SYNC0>;
```

(b) SPHW without Overlapping

```
LSU
START:
  while( MB[0] == 0 );
  MB[0] = MB[1] = 0;
Prologue:
  LLS(1,MB[2],MB[5],1,RLS);
  do{ LLS(1,MB[2],MB[5],1,RLS);
      SLS(1,MB[3],MB[5],1,WAIT);
  }while( MB[4] > MB[2] );
Epilogue:
  SLS(1,MB[3],MB[5],1,WAIT);
  MB[1] = 1;
  goto START;
```

```
EXU
<SYNC0> if( WAIT( ) ) goto <SYNC0>;
        else          goto <EXE0>;
<EXE0>  DBRO[0][WR+0]=DBRI[0][WR+0]
                     *DBRI[0][WR+0];
        goto <EXE1> ;
<EXE1>  DBRO[0][WR+1]=DBRI[0][WR+1]
                     *DBRI[0][WR+1];
        goto <EXE2> ;
<EXE2>  DBRO[0][WR+2]=DBRI[0][WR+2]
                     *DBRI[0][WR+1];
        goto <EXE3> ;
<EXE3>  DBRO[0][WR+3]=DBRI[0][WR+3]
                     *DBRI[0][WR+3];
        goto <SYNC1> ;
<SYNC1> RLS( ); WR += 4;
        goto <SYNC0> ;
```

(c) SPHW with Overlapping Using Software Pipelining

Fig. 4. Simple Mapping Example

We assume that the line contains 4 words and the data buffer registers have 1 bank. The number of entries of the bank is 4 in Fig. 4 (b) and 8 in Fig. 4 (c). In Fig. 4 (c), to implement the double buffer, the data buffer registers have two times capacity than that of Fig. 4 (b).

For mailboxes (MB[0]-MB[5]), the MB[0] and the MB[1] are the flags to invoke the SPHW and to inform of the completion respectively. The MB[2-5] hold the start address to be loaded, the start address to be stored, the end address to be loaded, and the stride width (16byte) respectively.

In Fig. 4 (b), the LSU loads 4 array elements into the DBRI[0][0-3] by the load line stream (LLS). At the same time, the LSU issues the release synchronization (RLS) as the producer to the EXU. Then, by the store line stream (SLS), the LSU attempts to store the output data buffer register (DBRO[0][0-3]) into the memory with the wait synchronization (WAIT). The EXU stays at the synchronization state ($\langle \mathrm{SYNC}_0 \rangle$) until the LSU prepares the data on the DBRI. Then, the LSU starts the execution and produces the processed data into the DBRO from $\langle \mathrm{EXE}_0 \rangle$ to $\langle \mathrm{EXE}_3 \rangle$. The EXU issues the release synchronization to the LSU at the $\langle \mathrm{SYNC}_1 \rangle$. The LSU invoked by the EXU can store the DBRO into the memory.

This straight-forward mapping mentioned above suffers from memory access latency. Although there may be several methods to hide memory access latency, in this example we show the simple method using the software pipelining [10]. In the software pipelining, the LLS and the SLS in the main loop are copied to the front of the main loop and the back of it respectively. In the main loop, the data used at the next iteration is loaded at the current iteration. Thus, the memory accesses of the LSU are overlapped with the data processing of the EXU.

3 Case Study for Evaluation

3.1 Prototype

To perform the preliminary evaluation, we have developed the prototype of the SPHW (SPHW-1). We have used the ISE9.2.04i and the Virtex4 FPGA (XC4VLX25-10). In the SPHW-1, the width of the registers is 32 bit except for the data buffer pointers (DBPI and DBPO). The width of the data buffer pointers is 16bit. The mailbox and the general purpose register contain 32 entries. The LSU is a simple scalar processor with the 3-stage pipeline. When the LSU branches, 3-clock cycles are required to fill the pipeline again. For the LSU program, we have developed the tool that converts the C-based behavioral description shown in Fig. 4 to the binary file.

The SPHW-1 uses one blockRAM for the LSUMEM. The mailbox (MB) and the general purpose register (GPR) are constructed by the distributed RAMs.

3.2 AES

The AES [7] cipher encrypts plaintext on a block-by-block basis. The block length is 128bit. In the main process, the block in the plaintext is loaded from the memory, the loaded block is encrypted, and the encrypted block is stored into the ciphertext in the memory. To this main looping process, the software pipelining is applied to hide memory access latency. Fig. 5 shows the execution snapshot.

The capacity of the DBRI and DBRO is doubled to implement a double buffer as shown in Fig. 5. At T$_0$, the EXU is stalled until the LSU loads the block 0 from the memory into the DBRI A in executing the prologue. At T$_1$, the EXU processes the block 0 on the DBRI A and outputs the ciphered block 0 to the DBRO A. At same time, the LSU loads the next block 1 from the memory into the DBRI B in executing the kernel. At T$_2$, while the EXU processes the block 1, the LSU stores the ciphered block 0 from the DBRO A into the memory and loads the next block 2 into DBRI A. Thus, the cipher process by the EXU and the memory access by the LSU are overlapped.

3.3 Window-Based Image Filter [13]

For the window-based image processing on the FPGA, the memory buffering methods that try to hide the memory access latency and reduce the redundant

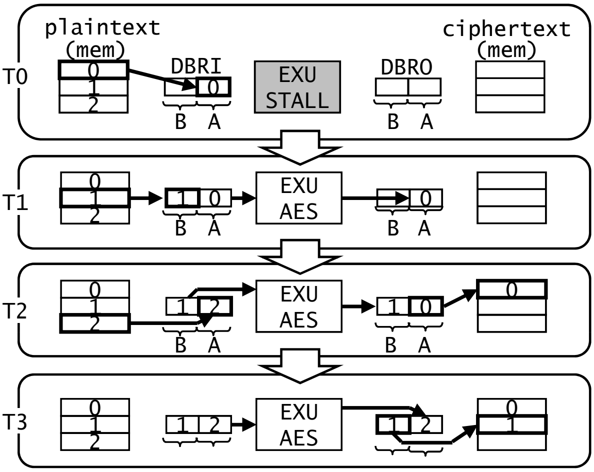
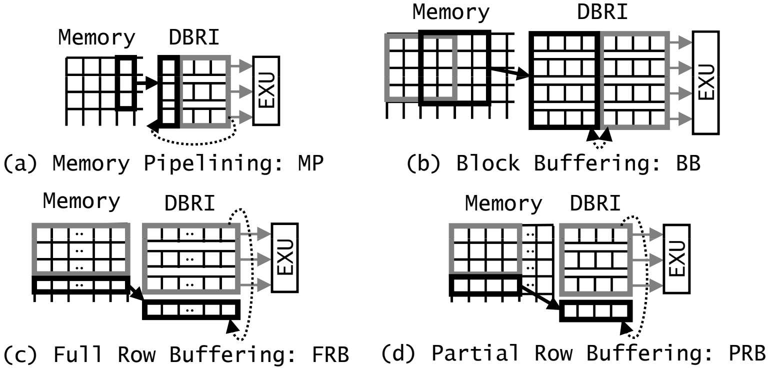


Fig. 5. Mapping of AES



(a) Memory Pipelining: MP

(b) Block Buffering: BB

(c) Full Row Buffering: FRB

(d) Partial Row Buffering: PRB

Fig. 6. Buffering Methods for Image Filter

memory accesses have been proposed [5,8,9,14]. By using the SPHW for different buffering methods, the implementation and the tradeoff of them can be performed efficiently. Also we show a new buffering method that suits to the SPHW architecture. Fig. 6 shows the conventional buffering methods and our method. We assume that the window size is 3×3.

Fig. 6 (a) is the memory pipelining (MP) [5,8]. The MP loads the vertical 3 pixels horizontally. The software pipelining (SP) is applied to the loop that proceeds the loading horizontally. To implement the double buffer, the input data buffer register (DBRI) of 3×4 is used as the ring register toward the entry direction. While the LSU loads the next vertical 3 pixels into the DBRI as the ring register, the EXU performs the 3×3 filter on the DBRI and moves the window to the entry direction.

Fig. 6 (b) is the block buffering (BB) [14]. The BB accesses the memory through the block larger than the window. The SP is applied to the loop that moves the block horizontally. To implement the double buffer, the DBRI has the capacity of 2 blocks. The LSU loads the next block into one buffer on the DBRI while the EXU performs the 3×3 filter on the other block.

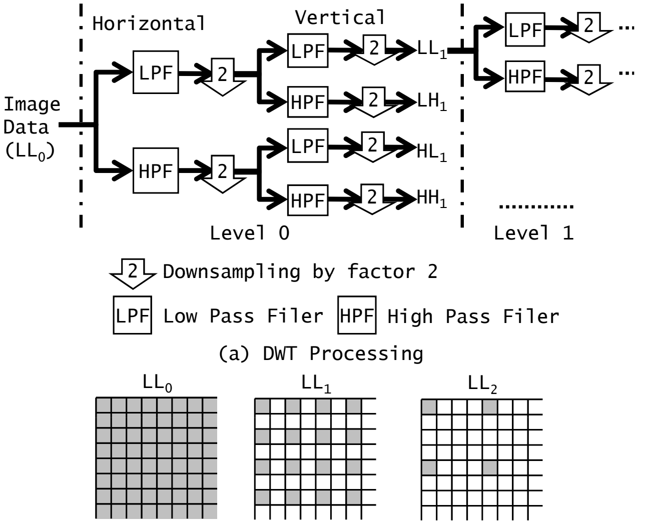
Fig. 6 (c) is the full row buffering (FRB) [5,9]. At first, the 3 rows from the top of the image data are loaded into the buffer. Then, the next row is loaded into the buffer. The SP is applied to the loop that loads the next row vertically. To implement the double buffer, the DBRI has 4 banks as the ring register toward the bank entry. The depth of the bank is equal to the width of the image data. The LSU loads the next row into the DBRI as the ring register while the EXU performs the 3×3 filter on the 3 rows in the DBRI.

The capacity of the data buffer registers is an important factor that decides the hardware size. In the FRB, the capacity of the DBRI and the DBRO is dependent on the image width. Thus, the buffering method of which the capacity is independent of the image data size is desired. Fig. 6 (d) shows the partial row buffering (PRB) that we have developed. The PRB is not dependent on the image data size for the capacity of the data buffer register. In the PRB, the partial row whose width is smaller than the image data width is loaded like the FRB. Once loading the partial row reaches the bottom of the image data, it moves horizontally and proceeds from the top of the image data similarly. The SP is applied to the loop that loads the next partial row. The LSU loads the next partial row into the DBRI as the ring register while the EXU performs the 3×3 filter on the 3 partial rows in the DBRI.

3.4 Discrete Wavelet Transform with 5/3 Filter [12]

The discrete wavelet transform (DWT) is one of the methods that analyze the frequency characteristic of a digital signal. Since the DWT suits compression applications for the image data well, JPEG2000 standard adopts it. In this paper, we focus on the DWT with the 5/3 filter [2], which is a mother wavelet used in the JPEG2000. The DWT with the 5/3 filter shows the different memory access patterns according to the processing phase. By mapping the SPHW on such application program, we can show a descriptive ability of the SPHW that can provide the uniform data processing for the EXU even if the various memory access patterns occur during execution.

Fig. 7 (a) shows an overview of the DWT processing [2]. The LPF means the low-pass filter. The HPF means high-pass filter. They construct the 5/3 filter as the mother function. ↓ 2 represents the downsampling with factor 2. The process that lies between the dash-dotted lines is called as the decomposition. As the level of decomposition proceeds, the number of data is reduced due to the downsampling. That is, the allocation of the data used is different across the levels of decomposition as shown in Fig. 7 (b). On the memory, the stride of elements in the LL_n (n is the level of decomposition) becomes wider as the level of decomposition proceeds.



(a) DWT Processing

(b) Memory Allocation of Array Elements to be Accessed

Fig. 7. DWT Processing and Memory Allocation

For the 5/3 filter, we used the following expressions for the LPF and HPF simplified by the lifting [2], which perform convolution on the image data.

$$Y(2n+1) = X(2n+1) - \left\lfloor \frac{X(2n) + X(2n+2)}{2} \right\rfloor \tag{1}$$

$$Y(2n) = X(2n) - \left\lfloor \frac{Y(2n-1) + Y(2n+1) + 2}{4} \right\rfloor \tag{2}$$

Eqs. (1) and (2) correspond to the HPF and LPF respectively. In each decomposition level, eqs.(1) and (2) are performed over each column of the image data and they are performed over each row. The software pipelining is applied to each loop processing the column and the row in order to hide the memory access latency.

Fig.8 shows the overview of mapping the DWT to the SPHW. At T_0, the EXU is stalled until the LSU loads first three pixels into the DBRI. At T_1, the EXU performs the 5/3 filter to the three pixels on the DBRI and the processed two pixels into the DBRO. At same time, the LSU loads the next two pixels into the DBRI as the ring register toward the bank direction. At T_2, the LSU stores the processed pixels on the DBRO into the memory and loads the next pixels in a ring register fashion. Simultaneously, the EXU processes the pixels on the DBRI as ring register. Henceforth, the data process by the EXU and the memory access by the LSU are overlapped until the end of the column or row.

As mentioned above, the memory allocation dynamically changes across the decomposition levels. However, from the viewpoint of the EXU, the data to be

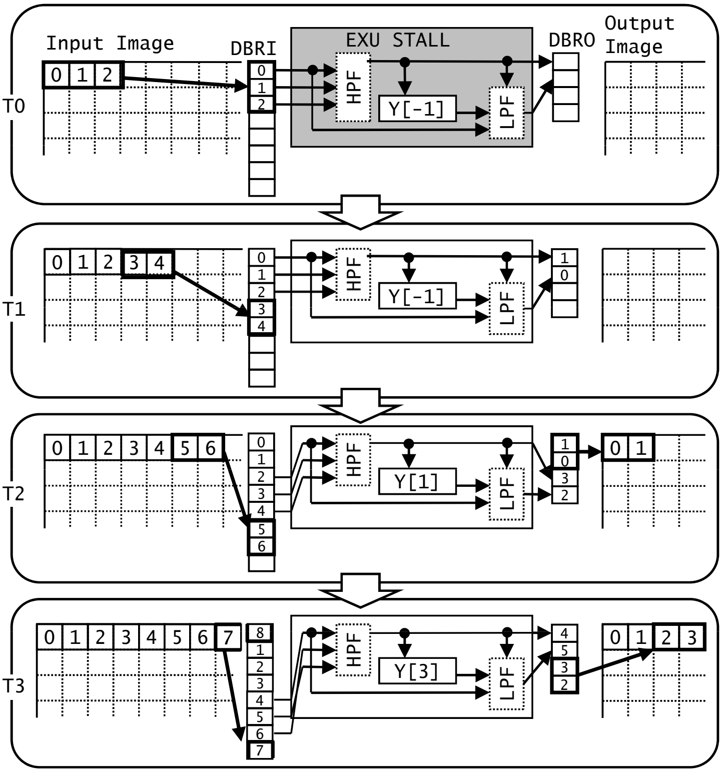


Fig. 8. Mapping DWT to SPHW

processed is just sequential data on the ring register of the DBRI/O. That is, you can design the uniform data processing hardware in spite of different memory access patterns.

4 Experiment and Discussion

4.1 Design Load and Hardware Overhead

To perform the comparative evaluation, we have developed the hardware whose memory access controller is written in the VHDL. This hardware with the custom memory controller is referred to as the CSTM. In the CSTM, the LSU in Fig.2 is replaced by the finite state machine optimized to the memory access pattern and the data prefetching. The data buffer register is constructed by the distributed RAMs and has the smallest capacity to implement the buffer used. We have no high-level synthesis tool supporting the EXU yet. So, the EXU is written in VHDL and designed as the optimum hardware for processing the data in the data buffer registers. In the SPHW-1 and the CSTM, we have used the same data

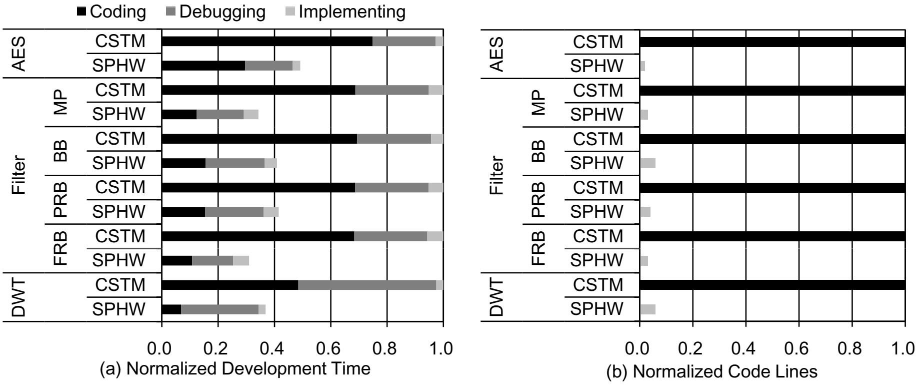


Fig. 9. Development Time and Code Lines

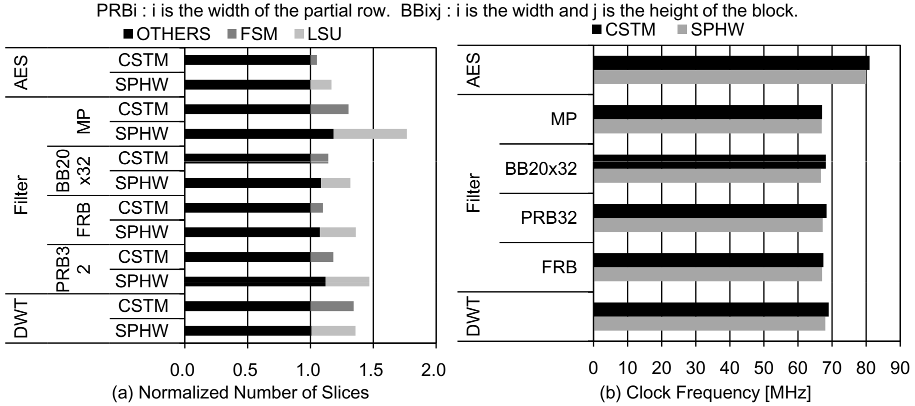


Fig. 10. Implementation Result

buffer registers and the EXU. This is because if the whole hardware is designed from scratch the smallest buffer and the suitable data processing hardware are likely to be equal to the data buffer registers and the EXU. In the CSTM, we have reduced the entries of the MB and GPR as much as possible.

Fig.9 (a) shows the coding time, the debugging time and the implementing time. As the reference, Fig.9 (b) shows the number of code lines. For Fig.9, the vertical axis is normalized to the CSTM in each program. Compared to the CSTM, the SPHW-1 reduces the coding time of 61% to 86%. This is because the SPHW-1 reduces the number of code lines from 94% to 98% compared with the CSTM. In addition, the SPHW-1 reduces the debugging time of 20% to 44%. Instead of revising the large VHDL program, revising the smaller LSU program leads to the reduction of the debugging time. The implementing time is the time consumed until the VHDL code is compiled and the bit stream is downloaded

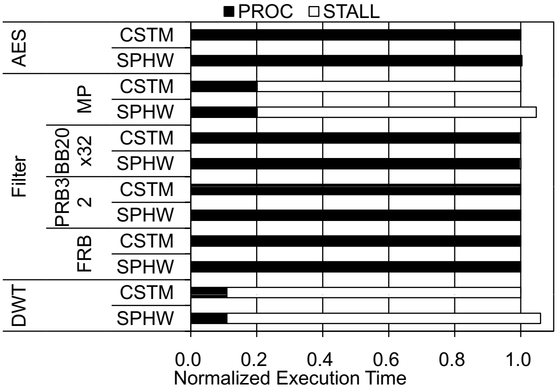


Fig. 11. Result of Performance Evaluation

into the FPGA. The implementing time is lower than 6% of the development time. Consequently, the SPHW-1 reduces the development time of 51% to 69%.

Fig. 10 shows the implementation result. The number of slices used for the LSU and the FSM in the CSTM is also shown as the breakdown. Both results of the maximum clock frequency reported by the ISE9.2.04i are comparable. This is because the critical path resides on the EXU. The propagation delay from the EXU is cut by the reconfigurable register file. For the slice, the CSTM requires from 183 to 592 slices while the LSU requires 603 slices. In addition, the SPHW-1 requires 161 slices for the MB and the GPR while the CSTM requires from 18 to 45 slices for them. So, the SPHW-1 consumes more slices than the CSTM.

Through this experiment, the result shows that the significant reduction of the development time by the SPHW can compensate the increased amount of hardware.

4.2 Performance

For the performance evaluation, we have used the ML401 board [11] with the Virtex4 FPGA (XC4VLX25-10). The on-chip clock frequency is 50MHz and the width of the on-chip bus is 64bit. The clock frequency for the DDR SDRAM is set to 100MHz and the burst length is set to 4. Before measuring, the data used is downloaded into the DDR SDRAM on the ML401. The image data size is 128×128. The pixel size is 32bit that contains 8bit R, G and B data. Before measuring, the image data is put into the DDR SDRAM.

Fig.11 shows the result of the performance evaluation. The vertical axis is normalized to the CSTM in each program. The PROC means the execution time of the EXU. The STALL indicates the stalled time of the EXU due to memory accesses. The performance differences between the SPHW-1 and the CSTM are about 6% in the MP and the DWT. In the other cases, the performance differences are lower than 0.1%. In the MP and the DWT, the processing time of the EXU is very small. Thus, the memory access latency is not enough hidden by the EXU processing. Consequently, the performance difference between the LSU and the FSM of the CSTM appears directly. Except for the MP and the DWT, the performance of the SPHW-1 and the CSTM is comparable because the memory access latency is enough hidden. Across the buffering methods the tendency of the SPHW-1 is similar to that of the CSTM.

Through these experiments, it is also confirmed that if memory access latency is enough hidden the SPHW-1 can be employed as a final product.

5 Conclusion

We have proposed a design-level hardware architecture (semi-programmable hardware: SPHW) which is inserted onto the pass from C to hardware. On the SPHW, the memory accesses and buffers are realized by the software programming and parameters respectively. By using the SPHW you can easily develop the data processing hardware containing the efficient memory access controller at C-level abstraction.

We have evaluated the SPHW using some application programs that show different memory access patterns. Compared with the case that the custom data prefetching circuit is attached instead of the LSU, the SPHW can reduce a design cost significantly by the LSU programming and achieve a comparable performance.

As future work, we will develop the HLS tool for the EXU using a conventional HLS tool. Then, we will perform the comparative evaluation between the C-based design environment of the SPHW and the conventional one that can generate the memory access controller.

Acknowledgement

This research is partially supported by the Kayamori Foundation of Informational Science Advancement.

References

1. Agility Design Solutions Inc.: Handel-C Language Reference Manual RM-1003-4.4. Agility (2007)
2. Angelopoulou, M.E., Masselos, K., Cheung, P.Y.K., Andreopoulos, Y.: Implementation and Comparison of the 5/3 lifting 2D Discrete Wavelet Transform Computation Schedules on FPGAs. Journal of Signal Processing Systems 51(1), 3–21 (2008)
3. Gupta, S., Savoiu, N., Dutt, N., Gupta, R., Nicolau, A.: Using global code motions to improve the quality of results for high-level synthesis. IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems 23(2), 302–312 (2004)
4. Lau, D., Pritchard, O., Molson, P.: Automated Generation of Hardware Accelerators with Direct Memory Access from ANSI/ISO Standard C Functions. In: IEEE Symp. on Field-Programmable Custom Computing Machines, pp. 45–56 (2006)

5. Liang, X., Jean, J.: Data buffering and allocation in mapping generalized template matching on reconfigurable systems. The Journal of Supercomputing 1(19), 77–91 (2001)
6. Mitrionics: Mitrion Users'Guide 1.5.0-001. Mitrionics (2008)
7. NIST: Federal Information Processing Standard Publication 197, Advanced encryption standard (AES) (2001), http://csrc.nist.gov/publications/fips/fips197/fips197.pdf
8. Park, J., Diniz, P.C.: Synthesis of pipelined memory access controllers for streamed data applications on FPGA-based computing engines. In: Proc. of Intl. Symp. on Systems Synthesis, October 2001, pp. 221–226 (2001)
9. Pellerin, D., Thibault, S.: Practical FPGA Programming in C. Prentice-Hall, Englewood Cliffs (2005)
10. Vanderwiel, S.P.: Data Prefetch Mechanisms. ACM Computing Surveys 32(2), 174–199 (2000)
11. Xilinx: ML401/ML402/ML403 Evaluation Platform User Guide. Xilinx (2006)
12. Yamawaki, A., Morita, K., Iwane, M.: An FPGA Implementation of a DWT with 5/3 Filter Using Semi-Programmable Hardware. In: Proc. of the Asia Pacific Conference on Circuits and Systems, pp. 709–712 (2008)
13. Yamawaki, A., Serikawa, S., Iwane, M.: An Efficient Comparative Evaluation to Buffering Methods for Window-based Image Processing Using Semi-programmable Hardware. In: Proc. of the International Conference on Engineering of Reconfigurable Systems & Algorithms, pp. 233–239 (2009)
14. Yu, H., Leeser, M.: Optimizing data intensive window-based image processing on reconfigurable hardware boards. In: Proc. of Workshop on Signal Processing Systems Design and Implementation, November 2005, pp. 491–496 (2005)

A New Approach: Component-Based Multi-physics Coupling through CCA-LISI

Fang Liu[1], Masha Sosonkina[1], and Randall Bramley[2]

[1] Scalable Computing Lab, USDOE Ames Laboratory
Wilhelm Hall 329, Ames, IA, U.S.A 50010
fangliu@scl.ameslab.gov, masha@scl.ameslab.gov
[2] Computer Science Department, Indiana University-Bloomington
Lindley Hall 215, 150 S. Woodlawn Ave. Bloomington, IN, U.S.A 47405
bramley@cs.indiana.edu

Abstract. A new problem in scientific computing is the merging of existing simulation models to create new, higher fidelity combined models. This has been a driving force in climate modeling for nearly a decade now, and fusion energy, space weather modeling are starting to integrate different sub-physics into a single model. Through component-based software engineering, an interface supporting this coupling process provides a way to invoke the sub-model through the common interface which the top model uses, then a coupled model turns into a higher level model. In addition to allowing applications to switch among linear solvers, a linear solver interface is also needed for the model coupling. A linear solver interface helps in creating solvers for the integrated multi-physics simulation that combines separate codes, and can use each code's native and specialized solver for the sub-problem corresponding to each physics sub-model. This paper presents a new approach on coupling multi-physics codes in terms of coupled solver, and shows the successful proof for coupled simulation through the implicit solve.

Keywords: Parallel model coupling; Component architecture and interface; Sparse matrix computation.

1 Introduction

Modeling physical phenomena with scientific computing is an interdisciplinary effort. Many problems in science and engineering are best simulated as a set of mutually interacting models; practical physical systems are often mathematically modeled by complicated Partial Differential Equations (PDEs). Many real-world systems involve a complex of multiple physical components. Scientists in many fields are becoming increasingly interested in coupling models together in order to advance their understanding. For example, the Sun-Earth system [29] presents a complex natural system of many different interconnecting elements: the solar wind transfers significant mass, momentum and energy to the magnetosphere, ionosphere, and upper atmosphere, and dramatically affects the physical processes in each of these physical domains. The Community Climate System Model (CCSM) [7] for global climate models comprises interdependent models that simulate the Earth's atmosphere, ocean, cryosphere, and biosphere.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 503–518, 2010.

These systems interact by exchanging energy, momentum, moisture, chemical fluxes, etc. For example, atmosphere provides to the Earth's surface downward radiative fluxes, momentum fluxes in the form of wind stress, and fresh water flux in the form of precipitation. Fusion energy simulation is now integrating codes that model different physics of a fusion reactor [11,20]. Most recently the Center for Simulation of RF Wave Interactions with Magnetohydrodynamics (CSWIM) [6] works on coupling existing codes to model the interaction between high power radio frequency (RF) electromagnetic waves, and magnetohydrodynamics (MHD) aspects of the burning plasma.

In various scientific simulation domains, a stand-alone model can run with simplified assumptions on the interaction of the particular domain with the rest of the system. Merging the existing simulation models to create a new higher fidelity combined model is a newly emerging theme in the scientific computing community. Data exchange based model coupling such as Model Coupling Toolkit (MCT) [22] and Earth System Modeling Framework (ESMF) [31] proves that models can be coupled through exchanging the boundary or initial condition between models. Models are generally mesh-based, time-stepping models and may be based upon highly complex or nonlinear algorithms and numerical schemes. The process of model coupling can occur on three different time scales: tightly coupled models operate at the fastest time scale where data are exchanged every time step; for slowly varying physics, the coupling needs only be done every few time steps; the slowest coupling operates once per simulation. In this paper, we won't get into the details of the above complexities, and mainly focus on the framework support in which models are to be embedded in order to form part of a coupled application.

Software component model for scientific computing may help in coupling the multiple physics codes, such that each code presents the clear interface for the center controller - **Coupler**. Our previous research work [23,24,27] designed a common interface LInear Solver Interface (LISI) which spans multiple high-performance computing (HPC) solver packages. Within Common Component Architecture (CCA) framework CCAFFEINE [8], each package can be encapsulated into a component providing standard interface so that easy switching of solver packages is achieved. In this paper, each sub-physics maps into an individual solver, a coupled physics becomes a coupled solver. It demonstrates the idea for coupling multi-physics codes through LISI interface by using implicit solve method. We first investigate some efforts on model coupling; then present the requirements for solver coupling; after analyzing the design in details, we provides a CCA coupling approach; finally we give a test case to validate the solver coupling idea.

2 Related Work

Some of coupling efforts have been done in the scientific computing community over the past 10 years. They have been deployed within their science domain [22,17,31,29,26,19]. None of them supports the wide variety of discretization schemes and numerical techniques of existing discretization schemes, and combining codes from different frameworks is still hard.

Tools such as MCT [22] that tries to provide a common utility to couple the climate models are mainly focusing on the data exchange between the models. The ESMF

[31] defines an architecture for composing complex, coupled modeling systems. The complicated applications are broken up into smaller pieces (components). Components are assembled together to create an application and the different implementations of component can be used in a plug-N-play (plug-N-play) fashion. The Space Weather Modeling Framework (SWMF) [29] aims at providing a flexible and extensible software architecture for multi-component physics-based space weather simulations. The SWMF uses layer architecture which is similar to the ESMF. A solid rocket motor simulation [19] requires consideration of multiple physical components, such as fluid dynamics, solid mechanics and combustion. Roccom provides an object-oriented software integration framework for inter-module data exchange and function invocation in parallel multi-physics simulations. The undergoing CSWIM [6] project is focusing on the interaction between high power radio frequency (RF) electromagnetic waves, and magnetohydrodynamics (MHD) aspects of burning plasma. It uses the batch management system and even system to couple the codes.

The above work demonstrates the successful usage in their perspective domains, but none of these frameworks has attracted a large user base or been widely adopted outside their field of application due to its lack of generality. CCA [14] has been started to address this problem, applying component principles at the level of whole applications, so that parallel applications can run both stand-alone and with other applications within the framework.

3 Solver Coupling Requirements

Coupled modeling is increasingly necessary to make progress in understanding the science of complex physical phenomena. The interaction in a combined simulation needs to be addressed by mathematical and physical aspects of simulation. Combined problem can be solved by forcing consistent solutions on the interfaces. The combination is not trivial because the constituent applications come with their own meshes, discretizations and internal data structures, especially in HPC, the data decomposition may be different.

Component models have been introduced to the module-coupling community recently, and already found themselves well suited for the requirements. The most promising approach is to define a standard set of interface functions that every physics component must provide. The interface describes the list of inputs and outputs, and exists independent of the implementation. Such common interfaces are present in some of the frameworks mentioned in section 2. Each sub-model needs to provide standard interface to the framework while the framework provides the data exchange interface for each sub-model to use.

In this section we analyze some of the challenges pertinent to the treatment of combined physics models as a coupled solver in a multi-physics simulation. The invocation of sub-model is through CCA LISI interface [23]. The target models include those that have

- an extensive software base of existing codes which were not designed to interoperate with other codes.

- an investment in the sub-model software which is too expensive to duplicate because of various reasons such as continuing active use, development, and evolution. In other words, it is not sufficient to use an earlier release or frozen version.
- an "awareness" of the other sub-physics, typically by a greatly simplified treatment of their contributions. For example, they assume that a boundary condition is not time-variant, or that some averaging is sufficient to reflect contributions from other sources, or that the constituent physics are not coupled.
- a vital need for inter-model interfaces, which requires active collaboration between application and computer scientists. The definition of those interfaces is ideally a community effort with broad intellectual support.
- parallelism requirement in the computations.

Through the participated models' feature, we may abstract the major requirements for the solver coupling:

- supporting multi-language legacy codes,
- providing data exchange sub-model running on the different size of processes (called MxN problem),
- requiring minimal changes for each physic model,
- allowing individual code evolvement,
- use of model's native and possibly highly specialized solvers.

In the following subsections, we will give detailed explanations on above requirements.

3.1 Multi-language Support

The large scale simulation codes always involve the modules developed by the different teams from different institutes. The code may be in different program languages, parallelism paradigms or platform dependability. The CCA [14] provides HPC language support, in particular, support for FORTRAN77, FORTRAN90, C, C++, Java, etc. This is achieved through Babel [16], a tool relying on the Scientific Interface Definition Language (SIDL) [21] to express software interfaces in a language neutral way. Babel compiler can generate the appropriate glue code stubs and skeletons based on SIDL to facilitate language interoperability. In turn, CCA supports the application to run with components of different languages.

3.2 Data Exchange

In a coupled model, the data exchanged by two components A_i and A_j reside on their overlap domain Ω_{ij}, and in principle each component will have its own discretization of Ω_{ij}. And two components may run on different numbers of processors, so that the data partitioning on the overlap domain may be different. This is so called M-by-N problem in the scientific coupling [12] community. In our design, we treat sub-models as in an algebraic way, such that we assume that some other components already did the data interpolation and M-by-N data exchange before the coupling interface is called. In our current research, we only target on the case when the coupled models run on the same number of processors, but in the future, when considering the other cases, we will take a look at how to integrate the M-by-N component with our coupling framework.

3.3 Minimal Code Changes

Existing software integration frameworks typically require large manual rewrites of existing codes, or specific tailoring of codes written to be used in the framework. The resulting special purpose code is not usable outside of the framework. The easy reuse of legacy software is one of effort of CCA [14], it minimizes the effort required to incorporate existing software into CCA environment. A thin layer of the component wrapper is added to the legency code, with well defined interface the legency code becomes a ready to use component. Since the target models are too expensive to rewrite, the above solution makes the minimal code change possible.

3.4 Individual Code Evolvement

The coupled model should allow each sub-model keep growing as long as the interface it provides is unchanged. In a coupled solver, each sub-model provides a LISI interface, only the adapter needs to be updated once the sub-model grows. The Component Based Software Engineering (CBSE) allows individual sub-model evolving without disturbing other participating sub-models.

3.5 Use of the Native Solvers

In a PDEs simulation, sometimes the general solver cannot fully solve the discretized problem due to complex geometry, and mesh shape, and discretized form of PDEs. The resulting linear system may have a very irregular sparsity pattern, large condition number, etc., which makes iterative method such as *GMRES*, *BICGSTAB* impossible to solve it. In this situation, it is hard to use the available solver packages. These simulations usually write their own solvers, in some way the solver is embodied with the simulation code. If this model is coupled with other models, the exposed LISI interface can make the model a special solver component.

While using CCA and LISI to define the coupled solver, the above requirements can be satisfied. And when each sub-physics is treated as a solver, the coupled solver idea can be used in multi-physics coupling.

4 Design of Coupled Solver

In order to analyze the coupled solver, some mathematics background needs to be demonstrated first. Domain decomposition method received a strong revival of interest in the end of 80s and early 90s due to its potential in parallel computing [15]. It is a class of techniques for the solution of PDEs on a domain by solving smaller problems on subdomains. They are particularly useful for solving problems on irregular domains and on parallel computers. The key ingredient is the system of equations governing the variables on the interface between the subdomains. The domain decomposition method idea can be traced back to Schwarz's alternating procedure, in which existence of solutions to boundary value problems are proved by an iteration involving solutions on overlapping subdomains. This idea is also widely used in many fields of scientific computing. Figure 1 shows these two ideas, and our current research borrows the domain decomposition idea in a reverse manner.

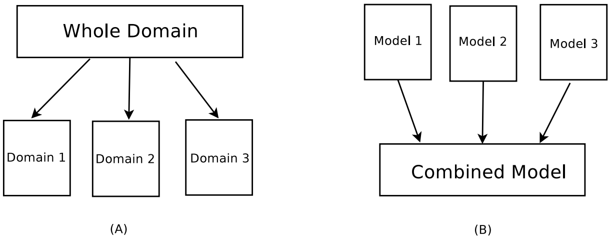


Fig. 1. (A) Domain decomposition used in parallel computing about 20 years ago; (B) Emerging modeling coupling from individual sub-model

4.1 Model Description

To simplify the illustration, we consider two-domain problem in Figure 2. There are two overlapping subdomains Ω_1 and Ω_2, the interface between two domains is named Γ.

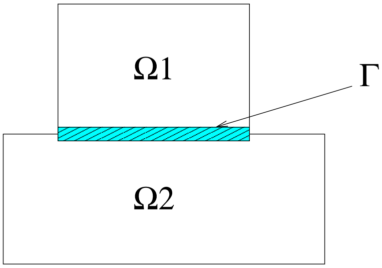


Fig. 2. Coupled domain

When the domain decomposition method idea is applied to multi-physics coupling, the domain $\Omega_1 and \Omega_2$ can be treated as the different physics simulations, which solve the different PDEs, for example. There are two cases for the problem to be solved

1. when there is no interaction between physics, the discretized system is solved as:

$$\begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{1}$$

   Here A_{11} and A_{22} are discretized form of each physics on its own domain, x_1 and x_2 are solution vectors for each physics, b_1 and b_2 are the right hand side vectors for each physics simulation.
2. When thinking about coupling two physics codes, a combined linear system is created in Figure 3. Besides A_{11} and A_{22} for each physics, a few new matrices are introduced through the interaction between two physics:
   - A_{33} represents the linear system for interface nodes, it is based on governing PDEs used on the interface nodes, it may have its own discretization scheme

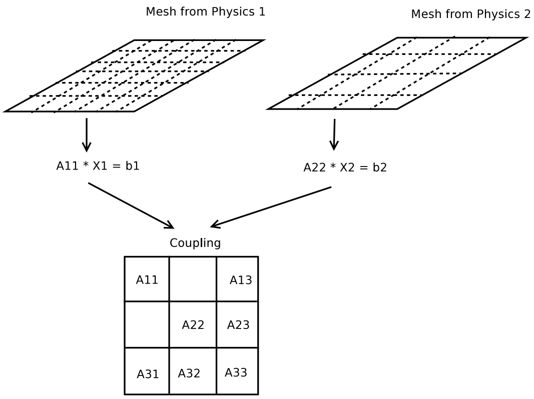


Fig. 3. Multi-Physics Coupling Diagram

used or use one of two coupled physics discretization scheme, which is the decision made by two physics simulation codes. Some data mapping needs to be done on these interface nodes. For example, the grid points of one simulation may correspond to the mesh of the coupled simulation. Usually data interpolation and MxN component need to solve this problem as we discuss in section 3.

- A_{13} and A_{13}^T are the coupling matrices between physics on Ω_1 and interface nodes, generally the interactions between them are symmetric from each side of point of view, so that the transpose of A_{13} represents the coupling from interface nodes to the nodes on the physics domain Ω_1.
- A_{23} and A_{23}^T are the coupling matrices between physics Ω_2 and interface nodes.
- x_3 and b_3 are the solution vector and right hand side vector for the interface nodes.

$$\begin{bmatrix} A_{11} & 0 & A_{13} \\ 0 & A_{22} & A_{23} \\ A_{13}^T & A_{23}^T & A_{33} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{2}$$

Through comparing non-coupling Equation 1 and coupling Equation 2, the system of linear equations becomes more complex to solve, especially when each sub-physics simulation runs in parallel.

4.2 Solver Methods

In this section, we are investigating two solver methods on solving Equation 2. Although both methods can be used, one is chosen over the other due to its guaranty on solver's convergence.

One of the most popular method on solving Equation 2 is the alternating Schwarz method [25]. The original alternating procedure described by Schwarz in 1870 consisted of three parts:

1. Alternating between two overlapping domains,
2. Solving the Dirichlet problem on one domain at each iteration,
3. Taking boundary conditions based on the most recent solution obtained from the other domain.

This procedure is called the Multiplicative Schwarz procedure [13,18]. This algorithm is therefore quite sequential, since each sub-domain updates its unknowns based on the other domain's previous step solution. It may not be the best candidate for parallel model coupling. The additive variant of the Schwarz procedure is more suitable for parallel processing, because the subdomains are not updated until a whole cycle of updates through all domains are completed [25], and the subdomains can be solved separately and at the same time.

When thinking coupled physics problem in terms of coupled solver, alternating Schwarz method is one of the options. But this method is slow and sometimes not convergent. There are two key assumptions associated with classical Schwarz method, and these assumptions are not verifiable from linear algebra arguments alone, see [25] (chapter 13.3). Given a linear system, it is unlikely that one can analytically establish that these assumptions are satisfied.

Another way to solve this problem is the Schur complement method [30]. The Schur complement arises naturally in solving (2) by using Block-Gaussian Elimination:

$$G = A_{33} - A_{13}^T * A_{11}^{-1} * A_{13} - A_{23}^T * A_{22}^{-1} * A_{23},$$

where G is the Schur complement system for solving the interface nodes, and G is much smaller than A_{22} and A_{33}. The solution of interface nodes then can be used to compute the solution for each physics subdomain. As we mentioned in section 1, we introduce the concept of **Coupler** to capture the interaction between two coupled physics codes. Assume that physics codes know nothing about the coupling matrix A_{13} and A_{23} which are only known to the coupler. **Coulpler** is an additional program doing extra work alongside two stand-alone physics simulations.

4.3 Coupled Solver Algorithm

In our approach, we chose the Schur complement method for our coupled solver, the reasons are at two folds: (1) Alternating Schwartz method is slow and sometimes not convergent. It can only couple the systems via their boundary conditions; (2) Schur complement method converges in fewer iterations for any coupled system and explicitly account for complex coupling interactions.

These solution processes can be done with Krylov subspace methods such as GMRES and BICGSTAB [10], which requires repeated matrix-vector multiplications. The algorithm 1 details the solution process.

Note that during each iteration, when Schur complement matrix is used, five matrix-vector products are called, and two linear solvers are deployed. The algorithm only requires the solution of problems with A_{11} and A_{22}, which corresponds to solving independent problems on the subdomains. This means that coupled solver can leave the subdomain solve unchanged while adding new functionality only in the coupler.

Algorithm 1. Coupling Algorithm through Schur-complement Method

```
Initialize:

  – subdomain $\Omega_1$ computes $b_1$ and $\omega_1 = A_{11}^{-1} * b_1$, return $\omega_1$ to coupler;
  – subdomain $\Omega_2$ computes $b_2$ and $\omega_2 = A_{22}^{-1} * b_2$, return $\omega_2$ to coupler;
  – coupler computes $b_3 = b_3 - A_{13}^T * A_{11}^{-1} * b_1 - A_{23}^T * A_{22}^{-1} * b_2$

Step 1: Solve $G * x_3 = b_3$: During each iteration, a matrix-vector product is conducted, ($x_{3p}$
represents the solution vector of $x_3$ from the previous iteration), and steps include:

  1. coupler computes $x_3 = A_{33} * x_{3p}$
  2. coupler computes $d_1 = A_{13} * x_{3p}$
  3. subdomain $\Omega_1$ computes $\omega_1 = A_{11}^{-1} * d_1$
  4. coupler computes $x_3 = x_3 - A_{13}^T * \omega_1$
  5. coupler computes $d_2 = A_{23} * x_{3p}$
  6. subdomain $\Omega_2$ computes $\omega_2 = A_{22}^{-1} * d_2$
  7. coupler computes $x_3 = x_3 - A_{23}^T * \omega_2$

Step 2: Compute

  – $x_1 = \omega_1 - A_{11}^{-1} * A_{13} * x_3$
  – $x_2 = \omega_2 - A_{22}^{-1} * A_{23} * x_3$
```

5 CCA Coupling Approach

Component-based software design combines object-oriented design with the powerful features of well-defined interfaces, programming language interoperability and dynamic composability [28]. While component-based design was initially motivated by the needs of business application developers, it also offers enormous potential benefits to the computational science community. The CCA [14] is a software component model that specially addresses HPC applications. In CCA, the software unit is treated as a component, each component connects to another component through a collection of public interface, or ports [1]. CCA employs a provides/uses paradigm in which a component provides a set of interfaces that other components can use.

In our previous research work, LISI [23,24,27] is an effort within CCA Forum to identify the common requirements among widely available HPC sparse linear solver packages, and abstract a common Application Programming Interface (API) that spans them. LISI is designed to facilitate the run time plug-N-play from multiple HPC solver packages. The generic solver interface *lisi.SparseSolver* is deployed as a CCA port which can be implemented by a solver component and used by another component. The auxiliary interface when a matrix free solver is used is also presented as *lisi.MatrixFree* interface. The component providing this interface will provide a matrix vector product functionality, and the component using this interface will be a solver component of solving a linear system in a matrix free manner. In order to support constructing a coupled solver for the multi-physics coupling simulation, the *lisi.BaseSolver* is abstracted from *lisi.SparseSolver* to support the matrix free solver. This interface only requires *uses* port to provide right hand side vector and get the solution back, and the block row partitioning is assumed in this interface.

LISI BaseSolver Interface

```
package lisi version 0.2
{
  interface BaseSolver extends gov.cca.Port
  {
    int initialize(in long comm);
    int setStartRow(in int startrow);
    int getStartRow();
    int setLocalRows(in int rows);
    int getLocalRows();
    int setupRHS(
      in rarray<double,1> RightHandSide(NumLocalRow),
      in int NumLocalRow);
    int solve(
      inout rarray<double,1> Solution(NumLocalRow),
      inout rarray<double,1> Status(StatusLength),
      in int NumLocalRow,
      in int StatusLength);

  }
}
```

5.1 Design Architecture

In section 4, we have analyzed the coupled solver in details from its mathematics aspect. From the algorithm we present in previous section, there must be at least three components for our CCA design, one for each subdomain physics simulation and one for the coupler. Since the coupler is also going to solve the Schur complement system for x_3, we introduce another solver component to solve the Schur complement system in a matrix free manner so that the Schur complement system does not have to be formed explicitly.

Next question we need to answer is what information each component should hold to best simulate the real world coupling situation? In our design we treat the coupled physics simulation as a coupled solver, for two physics subdomains, they should hold their own linear systems A_{11} and A_{22}, and right hand side vectors b_1 and b_2. Since physics simulation maintains its own application data, it runs as a stand alone application but now provides *lisi.BaseSolver* port. All the coupling information should retain within **Coupler** component, the information includes the coupling matrix between physics subdomain 1 with the interface nodes - A_{13}, and the coupling matrix between physics subdomain 2 with the interface nodes - A_{23}, and the discretization on the interface nodes with its own governing function - A_{33}. In the real world coupling simulation, these three matrices must be designed by application domain expert in order to construct a meaningful combined simulation.

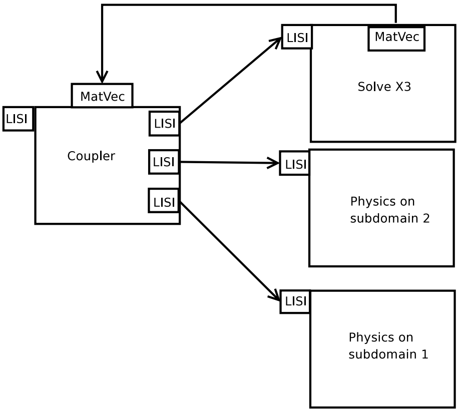


Fig. 4. Coupling With CCA Components

Figure 4 shows the design architecture for the coupled physics simulation, the component diagram follows the CCA uses-provides design pattern, the arrow means the calling direction.

- **Coupler** component has two provides ports: one is *lisi.BaseSolver* to provide the functionality to the outside application who may treat the coupled simulation as a solver; one is *Matvec* port which provides the matrix vector product for the Schur complement system when solving x_3. This component also has three uses ports, and all of them are *lisi.BaseSolver* ports. These ports are used to get the solution back from two physics simulation components and one solver component for x_3.
- **Physics** components on subdomain 1 and subdomain 2, they are basically converted from real world simulation codes to components through a provides port *lisi.BaseSolver*. Usually a thin layer is implemented on top of the legency codes. These two components are called once during each iteration in the coupling algorithm as descripted in Section 4.
- **Solver** component for x_3. It represents Step 1 in algorithm from Section 4. Since this component has to solve a Schur complement system in a matrix free fashion, it has a uses port of *MatVec* along with its provides port *lisi.BaseSolver*.

5.2 Implementation

This design makes **Coupler** as a central hub for the coupling, and **Physics** components and x_3_**Solver** component should run simultaneously to exploit concurrence to the most in the simulation overall. CCA is communication transparent specification, which makes it lightweight and simple to use [9]. In our design, we choose in the most intuitive way among several MPI constructions already known by application developers, such as MPI communicator groups. Components in the CCA coupling run in a single MPI instance (started by a single *mpirun* command). The MPI communicator world needs to be divided into several subgroups as follows:

- a1_group and a2_group are MPI communicator groups for Physics component on subdomain 1 and subdomain 2, respectively. Depending on how large each simulation is, these two groups differ in size. And they shouldn't overlap to each other, since two simulations need to run simultaneously.
- a3_group is for both Coupler component and x_3_Solver component. Since every time Coupler requires from x_3_Solver, it has to wait for the solution return in a blocking call manner. It is not crutial to have concurrency between these two components. Since the A_{33} is much smaller compared to A_{11} and A_{22}, the processes assigned to Coupler and x_3_Solver might have number fewer than the Physics component. And a3_group shouldn't share processes with either a1_group or a2_group.
- a13_group is the augmented MPI communicator group for both a1_group and a3_group, and a23_group is the augmented MPI communicator group for both a2_group and a3_group. These communicator groups are used for data passing between Coupler component and Physics components.

During initialization phase, two **Physics** components load their linear system A_{11} and A_{22}, and the right hand side vectors b_1 and b_2, act as a real application. **Coupler** component loads the coupling information A_{13}, A_{23}, A_{33} and b_3. Since all the components may run on different number of processors, data is partitioned into the different number of chunks. In our design, we tried to avoid **MxN** problem [12], so we make the physics components running on the same number of processors, and the coupler running on a single processor. But in the real coupling problem, when adding the **MxN** component, both coupler and physics components can run on any number of processors. In order to get the solution from subdomain 1 back to the coupler, solution is produced within MPI communicator group a1_group, but collected to coupler's processor through communicator group a13_group. The similar is true for the solution on subdomain 2.

During the solve phase, **Solver** component calls back through *MatVec* port during each of its iteration for solution of x_3. During each iteration step, solutions of A_{11} and A_{22} are needed from physics subdomain 1 or subdomain 2. The similar steps are applied as in initialization phase. One different thing is that now the right hand side vector needs to be sent from the coupler to each physics component, MPI scatter call is used within each augmented communicator groups. Also some synchronization should be done for signaling these solving processes such as a Boolean variable indicating the process starts. Basically physics components are waiting for the signal of *MatVec* from Coupler. Only if the signal is received, they participate the collective calls on *MPI_Scatterv* and *MPI_Gatherv*.

During the final computing phase, the newly computed solution from Schur complement system on x_3 is used to compute the solution vector for each physics subdomain as indicated in section 4.3. Now the coupling cycle is done, two physics problems are solved under the condition that they interact each other on the overlapped domain.

5.3 Combining Multiple Packages

In our implementation, we also want to demonstrate that multiple solver packages can be linked together to compose into a new higher fidelity solver. Since in our previous research [23], we have converted Trilinos AZTECOO [5], PETSc [3] and SuperLU

[4] into CCA components, they have provided *lisi.SparseSolver* port to other components to use. Now we need to reimplement those components with *lisi.BaseSolver* interface, and another new package is added - High Performance Preconditioners (HYPRE) [2]. HYPRE is a multigrid solver, and Hypre's BOOMERAMG is a algebraic multigrid solver. Our goal is to demonstrate the interface usage on coupling code, which allows each code to use its native (specialized) solver. We choose the AZTECOO solver for our physics simulation on subdomain 1, used the BOOMERAMG solver for our physics simulation on subdomain 2, and the PETSC solver as Schur complement solver for x_3. In this way, three widely used solver packages are deployed together in one application within a component based multi-physics simulation. Not only the idea of combining the multiple solver packages is validated, but also multi-physics coupling can be resolved.

6 A Test Case

In order to demonstrate the multi-physics coupling through the LISI interface, we build a prototype test problem which may arise in a typical real-world application scenarios. There are two domains both model the following 2-dimensional PDEs,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \tag{3}$$

with Dirichlet boundary conditions, discretized with five-point centered finite-difference scheme on $n_x \times n_y$ grid. Where $f = (2.0 - 6.0 * x - x^2) * sin(y)$ and boundary condition $b = x * x * sin(y)$. Figure 5 shows our test problem, domain 1 sits on the left side with domain boundary $[0, 1] \times [0, 1]$, domain 2 sits on the right side with $[1, 2] \times [0, 1]$. The interface nodes align vertically at $x = 1$ between two domains. And the interface domain is discretized with one dimensional PDEs. 400 discretized nodes are used along *x*- and *y*- direction, so the linear system order is about 160000. The same discretization is used for both domains. The test runs on the Linux cluster *Odin* in the Computer Science Department at Indiana University. *Odin* has 128 nodes, each with two dual core AMD Opteron 2.0 Ghz processor and 4GB RAM on each computing node.

Physics subdomain 1 runs Trilinos AZTECOO solver with maximum iteration number 500 and tolerance of $1.0e^{-6}$, the solver method BICGSTAB and preconditioner method Jacobi are used. Physics subdomain 2 runs HYPRE's BOOMERAMG solver with maximum iteration number 30, and tolerance of $1.0e^{-6}$, all other parameters are set to default. Schur complement system solver for x_3 runs Portable, Extensible Toolkit for Scientific Computation (PETSc)'s BICGSTAB method with maximum iteration number 500 and tolerance of $1.0e^{-6}$, since it uses PETSc matrix-free method, there is no preconditioner chosen. The test runs on 9 processes total, components for physics 1 and 2 each run on 4 processes, and coupler runs on 1 process. The test repeats for 10 times, and result is average of the runs. Figure 6 shows the history of residual from solving the Schur complement system on x_3, the residual decreases as the iteration number increases, and it converges at the iteration number 415. This result demonstrates that coupled system is successfully solved. Sub-physics on domain 1 solves in 286 iterations with Trilinos solver while sub-physics on domain 2 solves in 5 iterations with HYPRE solver.

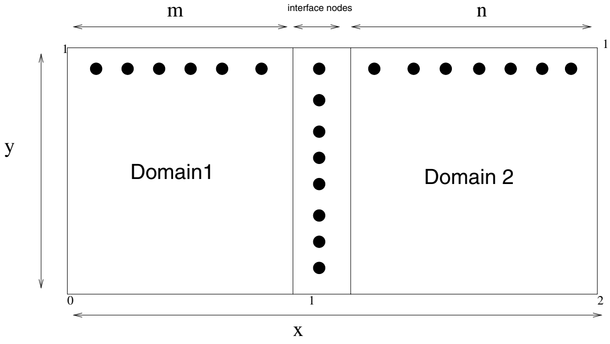


Fig. 5. Test problem setup

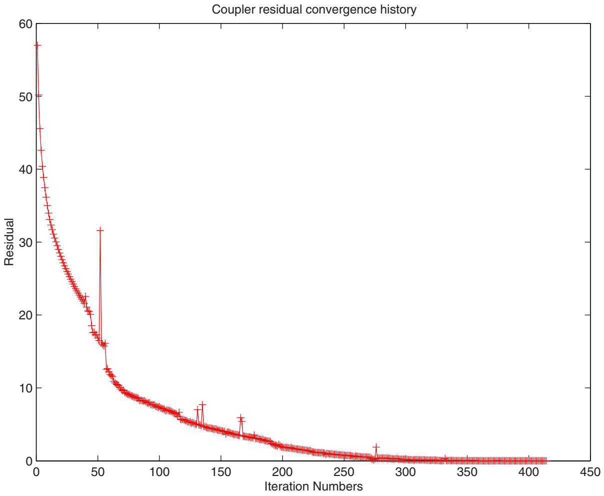


Fig. 6. Test result

7 Conclusion

In this paper, we demonstrate a new way of coupling multi-physics codes through CCA-LISI. This is the first model coupling approach within CCA, we generalize the model coupling idea through CBSE and treat coupling models as an abstracted solver. We outline a flexible approach to creating coupled model by introducing a new concept - coupled solver. When each sub-physics is treated as a sub-solve from the coupled physics, an implicit solve between multiple sub-physics can become a coupled solver. With CCA [14] technology, each sub-physics is encapsulated as a component with standard interface exposed to other components. It makes coupling easier through introducing extra component - **Coupler**. While each sub-physics model still runs separately, they all talk to **Coupler** to exchange the coupling information. The paper analyzes the requirements for a coupled solver, suggests two coupling algorithm and compares them. It also presents a coupling algorithm design along with its detailed implementation through CCA component framework; Although the tests were conducted with rather simple physics models, they nevertheless clearly show the coupled solver in action, and thus validate the proposed coupling of physics models under certain assumptions of data representation in algebraic form. As a future work, a generic way to represent the coupling information will need to be considered. For example, data interpolation needs to be done by other components when sub-models run on different numbers of processes. Being sufficiently general, the LISI interface, successfully used in this work, may be extended to incorporate new representations of the coupling information.

References

1. CCA specification (April 2008), http://www.cca-forum.org/specification
2. Hypre: Scalable Linear Solvers: high performance preconditioners (April 2008), http://www.llnl.gov/CASC/linear_solvers/
3. PETSc: Portable Extensibel Toolkit for Scientific Computation (April 2008), http://www-unix.mcs.anl.gov/petsc/petsc-as/
4. SuperLU: Direct Solver Package of large, sparse, nonsymmetric systems of linear equations (April 2008), http://crd.lbl.gov/~xiaoye/SuperLU/
5. The Trilinos Project (April 2008), http://software.sandia.gov/trilinos
6. Center for simulation of RF wave interactions with magnetohydrodynamics (March 2009), http://cswim.org
7. Community climate system model (March 2009), http://www.ccsm.ucar.edu/
8. Allan, B.A., Armstrong, R.: Ccaffeine Framework: Composing and Debugging Applications Iteratively and Running them Statically. In: Compframe 2005 workshop (June 2005)
9. Allan, B.A., Armstrong, R.C., Wolfe, A.P., Ray, J., Bernholdt, D.E., Kohl, J.A.: The CCA core specification in a distributed memory SPMD framework. Concurrency and Computation: Practice and Experience 14(5), 323–345 (2002)
10. Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., der Vorst, H.V.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd edn. SIAM, Philadelphia (1994)
11. Batchelor, D.: Integrated Simulation of Fusion Plasmas. Physics Today 58, 35–40 (2005)
12. Bertrand, F., Yuan, Y., Chiu, K., Bramley, R.: An approach to parallel MxN communication. In: Proceedings of the Los Alamos Computer Science Institute (LACSI) Symposium, Santa Fe, NM (October 2003)
13. Bjørstad, P.E.: Multiplicative and additive schwarz methods: Convergence in the two-domain case. In: Chan, T., Glowinski, R., Periaux, J., Widlund, O. (eds.) Domain Decomposition Methods, pp. 147–159. SIAM, Philadelphia (1989)
14. CCA-Forum. The DOE Common Component Architecture project (April 2008), http://www.cca-forum.org/
15. Chan, T.F., Resasco, D.C.: A Framework for the Analysis and Construction of Domain Decomposition Preconditioners. In: First International Symposium on Domain Decomposition Methods for Partial Differential Equations
16. Dahlgren, T., Epperly, T., Kumfert, G., Leek, J.: Babel User's Guide. CASC, Lawrence Livermore National Laboratory, Livermore, CA, babel-0.11.0 edition (2005)
17. Drummond, L.A., Demmel, J., Mechoso, C.R., Robinson, H., Sklower, K., Spahr, J.A.: A data broker for distributed computing environments. In: ICCS 2001: Proceedings of the International Conference on Computational Sciences-Part I, pp. 31–40. Springer, Heidelberg (2001)

18. Dryja, M.: An additive schwarz algorithm for two- and three- dimensional finite element elliptic problems. In: Chan, T., Glowinski, R., Periaux, J., Widlund, O. (eds.) Domain Decomposition Methods, pp. 168–172. SIAM, Philadelphia (1989)
19. Jiao, X., Campbell, M.T., Heath, M.T.: Occom: An object-oriented, data-centric software integration framework for multiphysics simulations. In: ICS, pp. 358–368 (2003)
20. Jill Dahlburg, J.C., et al.: Fusion Simulation Project: Integrated Simulation and Optimization of Magnetic Fusion Systems. Journal of Fusion Energy 20(4), 135–196 (2001), http://www.isofs.info
21. Kohn, S., Kumfert, G., Painter, J., Ribbens, C.: Divorcing Language Dependencies from a Scientific Software Library. In: 10th SIAM Conference on Parallel Processing. LLNL document UCRL-JC-140349, Portsmouth, VA, March 12-14 (2001), http://www.llnl.gov/CASC/components/babel.html
22. Larson, J., Jacob, R., Ong, E.: The model coupling toolkit: A new fortran90 toolkit for building multiphysics parallel coupled models. International Journal of High Performance Computing Application 19, 277–292 (2005)
23. Liu, F., Bramley, R.: CCA-LISI: On Designing a CCA Parallel Sparse Linear Solver Interface. In: Proc. 21th Int'l. Parallel & Distributed Processing Symp. (IPDPS), p. 10. ACM/IEEE Computer Society (2007)
24. Liu, F., Sosonkina, M., Bramley, R.: A HPC Sparse Solver Interface for Scalable Multilevel Methods. In: High Performance Computing Symposium, March 22-27 (2009)
25. Saad, Y.: Iterative Methods for Sparse Linear Systems. SIAM, Philadelphia (2003)
26. Shengsheng, Y., Yuanqiao, W., Liwen, H.,, D.: jian. Framework of Distributed Numerical Model Coupling System. In: DS-RT 2005: Proceedings of the 9th IEEE International Symposium on Distributed Simulation and Real-Time Applications, Washington, DC, USA, 2005, pp. 187–194. IEEE Computer Society Press, Los Alamitos (2005)
27. Sosonkina, M., Liu, F., Bramley, R.: Usability Levels for Sparse Linear Algebra Components. Concurrency and Computation: Practice and Experience 20, 1439–1454 (2008)
28. Szyperski, C.: Component Software: Beyong Object-Oriented Programming. ACM Press, New York (1999)
29. Toth, G., Volberg, O., et al.: A Physics-Based Software Framework for Sun-Earth Connection Modeling. In: Liu, A.T.Y., Kamide, Y., Consolini, G. (eds.) Multiscale Coupling of Sun-Earth Processes, Proceeding of the Conference on the Sun-Earth Connection, pp. 383–397 (2004)
30. Zhang, F.: The Schur Complement and Its Application. In: Numerical Methods and Algorithms., Springer, Heidelberg (2005)
31. Zhou, S.J.: Coupling climate models with the earth system modeling framework and the common component architecture. Concurr. Comput.: Pract. Exper. 18(2), 203–213 (2006)

Efficient Algorithms for the 2-Center Problems

Tadao Takaoka

Department of Computer Science
University of Canterbury
Christchurch, New Zealand

Abstract. This paper achieves $O(n^3 \log\log n/\log n)$ time for the 2-center problems on a directed graph with non-negative edge costs under the conventional RAM model where only arithmetic operations, branching operations, and random accessibility with $O(\log n)$ bits are allowed. Here n is the number of vertices. This is a slight improvement on the best known complexity of those problems, which is $O(n^3)$. We further show that when the graph is with unit edge costs, one of the 2-center problems can be solved in $O(n^{2.575})$ time.

1 Introduction

The k-center problem is important in network analysis in various areas, such as facility location in operations research, location of file servers in local area networks, analyzing influential persons in social networks, etc. By identifying k-centers, we can minimize the distance or time from those centers to members of the given network under various criteria.

In this paper we consider the 2-center problems for a directed graph, which is regarded as the special case of the k-center problems with $k = 2$. Given a directed graph with non-negative real numbers as edge costs, the k-center problems compute the set of k vertices, called centers, which serve the whole graph in an optimal way in two definitions. One is to minimize the longest distance from the set of centers to all vertices. Here the distance from the set to a vertex v is the shortest distance from the k vertices to v. We call this problem the absolute k-center problem. The other definition is to minimize the sum of the shortest distances to all vertices from the k vertices in the set. This latter definition by the sum, if divided by n, is regarded as the average distance from the set to all vertices, and thus called the average k-center problem. These two problems are defined in Kariv and Hakimi [11], [12] and Gary and Johnson [8], and shown to be NP-complete. When $k = 2$, straightforward algorithms of $O(n^3)$ time are known for the two 2-center problems. In [11] and [12] $O(n^2 \log n)$ time and $O(n^2)$ time algorithm for the absolute and average k-center problems are shown for trees. Frederickson [6] gives a linear time algorithm for the k-center problems for a tree with unit edge costs. Apart from trees, if we restrict the graph to a cactus, where cycles share at most one vertex, $O(n \log^2 n)$ time algorithm is known for the absolute k-center problem [3].

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 519–532, 2010.

We show in this paper those problems can be solved in $O(n^3 \log\log n/\log n)$ time for any directed graph after the all pairs shortest path (APSP) problem is solved in the same amount of time. We use the algorithm by the author [16], which is based on a fast algorithm for distance matrix multiplication (DMM). It is shown in page 204 of [1] that the time complexity of (n, n)-distance matrix multiplication (DMM) is asymptotically equal to that of the APSP problem for a graph with n vertices. Thus [16] mainly deals with DMM. The algorithm for DMM in [16] is based on the divide-and-conquer and table look-up approach.

Fredman [7] was the first to break the cubic bound $O(n^3)$ with $O(n^3(\log\log n/\log n)^{1/3})$ for the APSP problem. This complexity was improved to $O(n^3(\log\log n/\log n)^{1/2})$ by Takaoka [13] with RAM, and to $O(n^3/(\log n)^{1/2})$ by Dobosievicz [5] with extended logical operations. Since then, there have been some more progresses such as $O(n^3(\log\log n)/\log n)^{5/7})$ [9] and $O(n^3(\log\log n)^2/\log n)$ [15], and $O(n^3 \log\log n/\log n)$ [16]. Recently, algorithms with complexity $O(n^3\sqrt{\log\log n}/\log n)$ [18], $O(n^3(\log\log n/\log n)^{5/4})$ [10], $O(n^3(\log\log n)^3/\log^2 n))$ [4], etc., appeared.

Our algorithms for the 2-center problems are also based on matrix multiplication algorithms under different definitions, but the idea of using divide-and-conquer and table look-up is the same. The computational model in this paper is the conventional RAM, where only arithmetic operations, branching operations, and random accessibility with $O(\log n)$ bits are allowed all in $O(1)$ time. The basic idea in [7] and the subsequent improvements is that we speed up the computation by processing $O(\log n)$ bits in $O(1)$ time by either random accessibility with $O(\log n)$ bits or bit-wise logical operations on $\log n$ bits. We do not use bit-wise logical operations in this paper.

To multiply the small matrices resulting from dividing the original matrices, we sort distance data, merge sorted lists and use the ranks of those data in the merged lists. As the ranks are small integers, the multiplication can be done efficiently by looking at some pre-computed tables. We call this task of sorting "presort".

When edge costs are small integers, such as unit edge costs, we can solve the absolute 2-center problem in time more sub-cubic, that is, $O(n^{2.575})$. More precisely, $O(n^{2.575})$ is the time for solving the APSP problem with unit edge costs. Once the APSP problem is solved, the rest can be solved in $O(n^{2.376}\log n)$, where $O(n^{2.376})$ is the time for ordinary matrix multiplication. Thus we can say the APSP is the bottle neck for the absolute 2-center problem in this case.

The rest of the paper is as follows: Section 2 defines the two versions of k-center problems, and gives straightforward algorithms for them. Also the two versions of the 2-center problems are defined, and a formalization for solutions through matrix multiplication is given. In Section 3, we introduce the basic encoding scheme to deal with several small integers together in $O(1)$ time, which contributes to the speed-up of our algorithms. In Section 4, we review the divide-and-conquer algorithm for distance matrix multiplication given in [16] for two reasons. One is that the algorithm is used as the first stage of our 2-center algorithms. The other is that the technique used in [16], divide-and-conquer and table look-up for small matrices, is extended to our problems in this paper. In Section 5, new definitions of matrix multiplication are given, which are used for our 2-center problems. In Section 6, we show how to construct tables used in our algorithms, and show that the times for constructing those tables are within the claimed time complexity. Section 7 summarises the whole algorithms based on the parts described in the earlier sections. Section 8 is devoted to an efficient algorithm for the absolute 2-center problem with edge costs of small integers. Section 9 discusses a possible application of the 2-center algorithms for k-center problems, and concludes the paper.

2 *k*-Center Problems

Let $G = (V, E)$ be a directed graph with edge costs of non-negative real numbers. Vertices are given by integers between 1 and n, and edges are by pairs of vertices. Let $d(i, j)$ be the edge cost from vertex i to vertex j, and $d^*(i, j)$ be the edge cost of the shortest path from i to j. The shortest path from i to j is the path with the minimum sum of costs of edges over all possible paths. Let the matrices D and D^* be the matrices whose (i, j) elements are $d(i, j)$ and $d^*(i, j)$ respectively. The problem of computing D^* are known to be the all-pairs shortest path (APSP) problem and well studied. We use the algorithm in [16].

We define two kinds of k-center problems in this paper. Let C be a subset of V and $k = |C|$, the size of C. Let distance from C to vertex v, $dis(C, v)$, be defined by

$$dis(C, v) = \min\{d^*(u, v) | u \in C\}$$

If C is the set of fire stations, v is a house in a town, and edges are roads, $dis(C, v)$ is the distance from the nearest station to the house, although we deal with the more general model of directed graph. The cost of C is measured by the following two measures of $abs(C)$ and $ave(C)$.

$$abs(C) = \max\{dis(C, v) | v \in V\}$$
$$ave(C) = \Sigma_{v \in V} dis(C, v)$$

Finally the optimal k-center with absolute measure, C_{abs}, and that with average measure, C_{ave}, are defined by C that gives c_{abs} and c_{ave} in the following.

$$c_{abs} = \min\{abs(C) | C \subset V \& |C| = k\}$$
$$c_{ave} = \min\{ave(C) | C \subset V \& |C| = k\}$$

Intuitively speaking, C_{abs} is to minimize the longest distance from any fire station to houses while the number of stations is fixed to k. C_{ave} is to minimize the average distance under the same condition. The actual average is c_{ave}/n. In [12], C_{ave} is known as p-medians.

Both problems are known to be NP-complete, and thus there will unlikely be any polynomial time algorithm. The following is a straight-forward algorithm of exponential time based on a fast APSP algorithm. In the following consecutive statements in the same indentation level are regarded as being in the scope of the preceding "for".

ALGORITHM 1. *k-Center with absolute measure*

```
1. Solve the APSP problem by any fast algorithm
2. $opt\_value = \infty$
3. for $C \subset V$ such that $|C| = k$ do
4.     $abs = -\infty$
5.     for $v \in V$ do
6.         $dis = \infty$
7.         for $u \in C$ do $dis = min\{dis, d^*(u, v)\}$
8.         $abs = max\{abs, dis\}$
9.     $opt\_value = min\{opt\_value, abs\}$
10. $c_{abs} = opt\_value$
```

The algorithm for the average measure can be derived with slight modification in the following.

ALGORITHM 2. *k-Center with average measure*

```
1. Solve the APSP problem by any fast algorithm
2. $opt\_value = \infty$
3. for $C \subset V$ such that $|C| = k$ do
4.     $ave = 0$
5.     for $v \in V$ do
6.         $dis = \infty$
7.         for $u \in C$ do $dis = min\{dis, d^*(u, v)\}$
8.         $ave = ave + dis$
9.     $opt\_value = min\{opt\_value, ave\}$
10. $c_{ave} = opt\_value$
```

In both algorithms, line 3, 5, and 7 are iterated $O(n^k)$, n, and k times respectively, resulting in total time of $O(M(n) + n^k kn)$, where $M(n)$ is the time for distance matrix multiplication. There are several algorithms with $M(n)$ less than $O(n^3)$, that is, sub-cubic. When $k = 2$, the time becomes $O(n^3)$. Our purpose is to improve the second term of complexity to be sub-cubic, so that the total time becomes sub-cubic.

We reformulate the problems for $k = 2$. Let us start from C_{abs}. Let the set of centers be $C = \{i, j\}$ and D' be the transposed matrix of D^*, that is, $d'(i, j) = d^*(j, i)$. We compute $abs(i, j)$ for all i and j by

$$abs(i, j) = \max_{k=1,n}\{\min\{d^*(i, k), d'(k, j)\}\}$$

Then c_{abs} can be computed in $O(n^2)$ time by

$$c_{abs} = min_{i=1,n;j=1,n}\{abs(i,j)\}$$

Let us define the (max, min)-product of matrices A and B by $P = AB$, where $P = \{p(i,j)\}$ is given by

$$p(i,j) = max_{k=1,n}\{min\{a(i,k), b(k,j)\}\}.$$

The set of $\{abs(i,j)\}$ for all i and j is given by D^*D' under (max, min)-matrix multiplication, abbreviated as (max, min)-multiplication. Note that if A and B are 0-1 matrices, this becomes Boolean matrix multiplication.

Similarly the 2-center with average measure can be reformulated. Let

$$ave(i,j) = \Sigma_{k=1,n}\{\min\{d^*(i,k), d'(k,j)\}\}$$

Then c_{ave} can be computed in $O(n^2)$ time by

$$c_{ave} = \min_{i=1,n;j=1,n}\{ave(i,j)\}$$

Let us define the (Σ, min)-product of matrices A and B by $Q = AB$, where $Q = \{q(i,j)\}$ is given by

$$q(i,j) = \Sigma_{k=1,n}\{min\{a(i,k), b(k,j)\}\}.$$

Then the set $\{ave(i,j)\}$ for all i and j is given by D^*D' under (Σ, min)-matrix multiplication, abbreviated as (Σ, min)-multiplication.

Thus our problem reduces to how fast we can compute the (max, min)- product and (Σ, min)-product for the given two matrices with non-negative real elements.

3 How to Encode a Short List of Small Integers

A short list of small integers bounded by μ, $\mathbf{x} = (x_1, x_2, ..., x_m)$, is encoded into a single integer $h(\mathbf{x})$ with $0 \leq x_i \leq \mu - 1$ for all i by

$$h(\mathbf{x}) = (x_1 - 1)\mu^{m-1} + ... + (x_{m-1} - 1)\mu + x_m - 1$$

Note that this function h is one-to-one, and those variable involved have values small enough that $h(\mathbf{x})$ is contained in a single word. The encoding can be done in $O(m)$ time by the Horner algorithm and decoding can be done in $O(m)$ time by successive division by μ. We use this encoding scheme with various variables in later sections. The maximum value of $h(\mathbf{x})$ is bounded by μ^m. Since $\mu^m = c^{m \log \mu}$ for some constant $c > 0$, we can choose m and μ such that $m = O(\log n / \log\log n)$ and $\mu = O(\log n)$ to satisfy $h(\mathbf{x}) = O(n)$. We use the same name c for various constants hereafter.

Let us call h the packing function, since the encoding is like packing small integers into a single word.

4 Brief Review of Distance Matrix Multiplication

The normal product of two matrices A and B, $C = AB$, is defined by

$$c_{ij} = \Sigma_{k=1}^{n} a_{ik} \cdot b_{kj}, (i, j = 1, ..., n) \qquad (1)$$

Now we use the divide-and-conquer approach, that is, divide A, B, and C into (m, m)-submatrices for $N = n/m$ as follows:

$$\begin{pmatrix} A_{1,1} & ... & A_{1,N} \\ ... & & \\ A_{N,1} & ... & A_{N,N} \end{pmatrix} \begin{pmatrix} B_{1,1} & ... & B_{1,N} \\ ... & & \\ B_{N,1} & ... & B_{N,N} \end{pmatrix} = \begin{pmatrix} C_{1,1} & ... & C_{1,N} \\ ... & & \\ C_{N,1} & ... & C_{N,N} \end{pmatrix}$$

Matrix C can be computed by

$$C_{ij} = \Sigma_{k=1}^{N} \{A_{ik} B_{kj}\} (i, j = 1, ...N) \qquad (2)$$

where the product of submatrices is defined similarly to (1). This divide-and-conquer approach is made possible thanks to the associative property of the Σ operation.

The distance matrix multiplication is to compute the following distance product $C = AB$ for two (n, n)-matrices $A = [a_{ij}]$ and $B = [b_{ij}]$ whose elements are real numbers. In the following we replace the addition and multiplication by various operations in such a way that we can still take the divide-and-conquer approach. The first is *min* for Σ and $+$ for "$\cdot$" given as follows:

$c_{ij} = \min_{k=1}^{n} \{a_{ik} + b_{kj}\}, (i, j = 1, ..., n)$ (3)

$C_{ij} = \min_{k=1}^{N} \{A_{ik} B_{kj}\} (i, j = 1, ...N)$ (4) (Each multiplication is similar to (3))

The "min" operation is defined on submatrices by taking the "min" operation componentwise. This product is called distance product or $(min, +)$-product. We have N^3 multiplications of distance matrices in (4). Let us assume that each multiplication of (m, m)-submatrices can be done in $T(m)$ computing time, assuming pre-computed tables are available. The time for constructing the tables is reasonable when m is small. The time for *min* operations in (4) is $O(n^3/m)$ in total. Thus the total time excluding table construction is given by $O(n^3/m + (n/m)^3 T(m))$. By choosing an appropriate value for m, we can show the time for $(min, +)$-multiplication is $O(n^3 \log\log n / \log n)$. More details of the algorithm for $(min, +)$-multiplication is given in Appendix.

5 (max, min)-Multiplication and (Σ, min)-Multiplication

By replacing the Σ and $\cdot$ operations pair by (max, min) and (Σ, min), we define the following two more matrix products. We call those products (max, min)-product and (Σ, min)-product. The two products, the main theme of this paper, are defined by

$c_{ij} = \max_{k=1}^{n}\{\min\{a_{ik}, b_{kj}\}\}, (i, j = 1, ..., n)$ (5) ((max, min)-product)

$C_{ij} = \max_{k=1}^{N}\{A_{ik}B_{kj}\}(i, j = 1, ...N)$ (6) (Each multiplication is similar to (5))

$c_{ij} = \Sigma_{k=1}^{n}\{\min\{a_{ik}, b_{kj}\}\}, (i, j = 1, ..., n)$ (7) ((Σ, min)-product)

$C_{ij} = \Sigma_{k=1}^{N}\{A_{ik}B_{kj}\}(i, j = 1, ...N)$ (8) (Each multiplication is similar to (7))

Let us rename A_{ik} and B_{kj} in (6) by A and B. Let $M = \{1, ..., m\}$. Let $S(i, j)$ and $T(i, j)$ be defined by

$S(i, j) = \{k|a_{ik} \le b_{kj}\}$, $T(i, j) = \{k|a_{ik} > b_{kj}\}$

Note that $S(i, j) \cup T(i, j) = M$. Utilizing the commutative property of max, we observe

$$\max_{k=1,n}\{\min\{a_{ik}, b_{kj}\}\} = \max\{\max_{k\in S(i,j)}\{a_{ik}\}, \max_{k\in T(i,j)}\{b_{kj}\}\} \quad (9)$$

Let us assume sorted lists of the k-th column of A and the k-th row of B are available. The sorted lists are denoted by E_k and F_k. Let the merged list of E_k and F_k be G_k. Let H_k and L_k be the lists of ranks of elements of the k-th column of A and k-th row of B in G_k.

Then we have

$G_k(H_k(i)) = a_{ik}$, $G_k(L_k(j)) = b_{kj}$

Let a binary vector $\mathbf{x}_{ij}$ be defined by $\mathbf{x}_{ij}[k] = 1$, if $a_{ik} \le b_{kj}$, and 0, otherwise. Here $\mathbf{x}[k]$ is the k-th element of vector $\mathbf{x}$. The vectors $\mathbf{x}_{ij}$ and its complement $\mathbf{x}'_{ij}$ are membership vectors of $S(i, j)$ and $T(i, j)$. We express this fact by $S(i, j) = set(\mathbf{x}_{ij})$ and $T(i, j) = set(\mathbf{x}'_{ij})$.

Let pre-computed tables MAX_i^a and MAX_j^b be available, which are defined by

$$MAX_i^a(h(\mathbf{x})) = \max_{k\in set(\mathbf{x})}\{a_{ik}\} \quad (10)$$

$$MAX_j^b(h(\mathbf{x})) = \max_{k\in set(\mathbf{x})}\{b_{kj}\} \quad (11)$$

We compute MAX_i^a and MAX_j^b for all possible $\mathbf{x}$, so that we can use the table for each $\mathbf{x}_{ij}$ and $\mathbf{x}'_{ij}$. Note that those tables can be computed on A and B separately.

Also we assume that for lists of integers $H = (H_1, ..., H_m)$ and $L = (L_1, ..., L_m)$ a pre-computed table $MAP(h(H), h(L))$, defined in the following, is available.

$$MAP(h(H), h(L)) = h(\mathbf{x}), \text{ where } \mathbf{x}[k] = 1, \text{ if } H_k \leq L_k, \text{ and } 0, \text{ otherwise.}$$

This table will be used for $H(i) = (H_1(i), ..., H_m(i))$ and $L(j) = (L_1(j), ..., L_m(j))$ for each (i, j), where $H(i)$ and $L(j)$ are substituted for H and L.

Since $a_{ik} \leq b_{kj} \Longleftrightarrow H_k(i) \leq L_k(j)$, we can compute (9) by looking up those tables in $O(1)$ time for each i and j in the following, where MAP' is the bit-wise complement of MAP.

$$c_{ij} = max_{k=1,n}\{\min\{a_{ik}, b_{kj}\}\} = \max\{\max_{k \in set(\mathbf{x}_{ij})}\{a_{ik}\}, \max_{k \in set(\mathbf{x}'_{ij})}\{b_{kj}\}\}$$

$$= \max\{MAX_i^a(h(\mathbf{x}_{ij})), MAX_j^b(h(\mathbf{x}'_{ij}))\}$$

$$= \max\{MAX_i^a(MAP(h(H(i)), h(L(j)))), MAX_j^b(MAP'(h(H(i)), h(L(j))))\}$$

Note that the direction of scanning for packing of ranks for $h(H(i))$ and $h(L(j))$ is orthogonal to the direction of scanning for merging the lists E_k and F_k and computing H_k and L_k. See Figure 1.

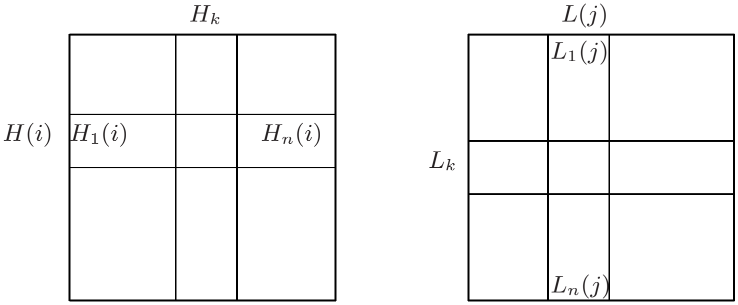


Fig. 1. H_k is column k in the left and L_k is row k in the right

$H(i)$ is row i in the left and $L(j)$ is column j in the right

Example 1. We consider (4, 4)-matrices. Let the i-th row of A be $\mathbf{a}_i = (12, 7, 4, 23)$ and the j-th column of B be $\mathbf{b}_j = (15, 6, 11, 20)$. Let $H(i) = (3, 2, 1, 4)$ and $L(j) = (4, 1, 3, 3)$. These numbers in $H(i)$ and $L(j)$ are determined by the relative order with other rows and columns. From this, we see $\mathbf{x}_{ij} = (1, 0, 1, 0)$

and $\mathbf{x}'_{ij} = (0,1,0,1)$. $\max\{MAX_i^a(h(1,0,1,0)), MAX_j^b(h(0,1,0,1))\} = \max\{\max\{12,4\}, \max\{6,20\}\} = 20$. Note that $\mathbf{a}_i$ and $\mathbf{b}_j$ can contain large numbers or real numbers of any precision, whereas H_i and L_j contain only numbers between 1 and 4, and $\mathbf{x}_{ij}$ is a binary vector.

Based on the method described, we now summarise our algorithm for (max, min)-multiplication in the following. Table construction and presort are mentioned in the upper structure of the algorithm in the following sections.

```
ALGORITHM 3. Compute $(max, min)$-product $C = AB$
1. Merge $E_k$ and $F_k$ to form $G_k$ for $k = 1, ..., m$
2. Compute $H_k$ and $L_k$ for $k = 1, ..., m$
3. Compute $h(H(i))$ for $i = 1, ..., m$
4. Compute $h(L(j))$ for $j = 1, ..., m$
5. Compute $c_{ij} = \max\{MAX_i^a(MAP(h(H(i)), h(L(j)))),$
                  $MAX_j^b(MAP'(h(H(i)), h(L(j))))\}$ for $i, j = 1, ..., m$
```

The algorithm for the average k-center is very similar. The equations (10) and (11) are replaced by

$$SUM_i^a(h(\mathbf{x})) = \Sigma_{k \in set(\mathbf{x})}\{a_{ik}\} \tag{12}$$

$$SUM_j^b(h(\mathbf{x})) = \Sigma_{k \in set(\mathbf{x})}\{b_{kj}\} \tag{13}$$

Using those mappings, we have the following algorithm.

```
ALGORITHM 4. Compute $(\Sigma, min)$-product $C = AB$
1. Merge $E_k$ and $F_k$ to form $G_k$ for $k = 1, ..., m$
2. Compute $H_k$ and $L_k$ for $k = 1, ..., m$
3. Compute $h(H(i))$ for $i = 1, ..., m$
4. Compute $h(L(j))$ for $j = 1, ..., m$
5. Compute $c_{ij} = \max\{SUM_i^a(MAP(h(H(i)), h(L(j)))),$
                  $SUM_j^b(MAP'(h(H(i)), h(L(j))))\}$ for $i, j = 1, ..., m$
```

We note that both Algorithms 3 and 4 take $O(m^2)$ time.

6 How to Construct the Tables

In Section 3 we used h for encoding small integers. In this section, we describe h more specifically as well as how to construct tables. We choose the value of m so as to satisfy the time for table construction is manageable. For μ, we have the following two choices. When we encode ranks in $H(i)$ or $L(j)$, we set $\mu = 2m$. When we encode a binary vector of size m, we set $\mu = 2$. Let $m = O(\log n / \log\log n)$. Then the size of the table is $O(n)$ or less by choosing an appropriate value for constant factor c such that $m \le c \log n / \log\log n$, as shown below.

To compute $MAP(\alpha, \beta) = h(x_1, x_2, ..., x_m)$ for arbitrary α and β, we decode α and β in $O(m)$ time, then compare the decoded lists of α and β, which are regarded as H and L in the previous section, one by one to get $x_1, ..., x_m$, and finally encode $x_1, ..., x_m$, spending $O(m)$ time. We do this for all possible α and β. The possible range of α and β is up to $(2m)^m$.

The total time for computing this table is thus $O(((2m)^m)^2 m)$. Observe

$$O(((2m)^m)^2 m) = O(c^{m \log m}) = O(n), \text{ for some constant } c > 0.$$

We note that table MAP is pre-computed, that is, computed independent of the input distance matrices, whereas tables MAX_i^a and MAX_j^b are computed based on the input matrices.

Now let us compute tables MAX_i^a and MAX_j^b. To compute $MAX_i^a(\alpha)$, we decode $\alpha = h(x_1, ..., x_m)$. Then take the maximum of $\{a_{ik}\}$ such that $x_k = 1$. MAX_j^b is computed similarly. Thus the time for MAX_i^a for all $i = 1, ..., m$ and MAX_j^b for all $j = 1, ..., m$ is $O(2^m m^2)$. Let us denote the collection of $\{MAX_i^a | i = 1, ..., m\}$ and $\{MAX_j^b | j = 1, ..., m\}$ by MAX^a and MAX^b. We need to construct those tables for N^2 sub-matrices given by A_{ik} and B_{kj} in (6). The time for those N^2 collections of tables is $O(N^2 m^2 2^m) = O(n^2 2^m) = O(n^2 n^{c/\log\log n}) = O(n^{2+\epsilon})$ for any $\epsilon > 0$.

The computation of tables SUM_i^a and SUM_j^b is similar.

7 Algorithm for the Whole Problem and Analysis

We summarise our algorithms for the absolute 2-center problem and average 2-center problem in the following. Note that we can use MAP, and other parts in common in the following two algorithms.

```
ALGORITHM 5. Absolute 2-center
1. Construct table $MAP$
2. Divide matrices $A$ and $B$ into $A_{ij}$ and $B_{ij}$ for $i, j = 1, ..., N$
3. Construct tables in $MAX^a$ and $MAX^b$ for $A_{ij}$ and $B_{ij}$ $(i, j = 1, ..., N)$
4. Sort $m$ columns of $A_{ij}$ and $m$ rows of $B_{ij}$ for $i, j = 1, ..., N$. // Presort
5. Compute $A_{ik}B_{kj}$ for $i, j = 1, ..., N$, by Algorithm 3
6. Compute $C_{ij} = \max_k\{A_{ik}B_{kj}\}$ for $i, j = 1, ..., N$
7. Compute the minimum element of matrix $C = \{C_{ij}\}$
```

```
ALGORITHM 6. Average 2-center
1. Construct table $MAP$
2. Divide matrices $A$ and $B$ into $A_{ij}$ and $B_{ij}$ for $i, j = 1, ..., N$
3. Construct tables in $SUM^a$ and $SUM^b$ for $A_{ij}$ and $B_{ij}$ $(i, j = 1, ..., N)$
4. Sort $m$ columns of $A_{ij}$ and $m$ rows of $B_{ij}$ for $i, j = 1, ..., N$. // Presort
5. Compute $A_{ik}B_{kj}$ for $i, j = 1, ..., N$, by Algorithm 4
6. Compute $C_{ij} = \Sigma_k A_{ik}B_{kj}$ for $i, j = 1, ..., N$
7. Compute the minimum element of matrix $C = \{C_{ij}\}$
```

As the above two algorithms are very similar, we analyze computing time for both algorithms together. Line 1 takes $O(n)$ time. Line 2 takes $O(n^2)$ time. Line 3 takes $O(n^{2+\epsilon})$ time. Sorting m columns of A_{ij} and m rows of B_{ij} takes $O(m^2 \log m)$ time. Thus line 4 takes $O(N^2 m^2 \log m) = O(n^2 \log\log n)$ time. Since computing $A_{ik}B_{kj}$ takes $O(m^2)$ time, line 5 takes $O(N^3 m^2) = O(n^3 \log\log n/\log n)$ time. Line 6 takes $O(n^3/m) = O(n^3 \log\log n/\log n)$ time. Line 7 takes $O(n^2)$ time. Thus in total these algorithms take $O(n^3 \log\log n/\log n)$ time.

8 When Edge Costs Are Small Integers

If edge costs are small non-negative integers, the complexity for APSP becomes deeply sub-cubic, i.e., $O(n^{3-\epsilon})$ for some $\epsilon > 0$, as shown in [14], [2], [17] and [19]. It is interesting to investigate whether we can use those sub-cubic algorithms for the APSP problem for the 2-center problems. Once the APSP problem is solved, the values in matrix D^*, the all-pairs shortest distance matrix, are no-longer small integers; they can be $O(n)$, even if edge costs are all one. Thus we cannot extend the technique used for the APSP problem to the 2-center problems straight away. We can efficiently solve the absolute problem in such a case by binary search as follows.

Let us assume the APSP problem for the given graph with unit edge costs has been solved with the shortest distance from vertex i to vertex j being $d^*[i,j]$. Let the threshold value t be initialized to $n/2$. Let a Boolean matrix B be defined by its element $b[i,j]$ as follows: $b[i,j] = 1$, if $d^*[i,j] \geq t$, and 0, otherwise. Let us square B to get the matrix $C = B^2$. From the equation $c[i,j] = \Sigma_{k=1}^{n} b[i,k]b[k,j]$, we observe that $c[i,j] = 1$ if and only if $b[i,k] = 1$ and $b[k,j] = 1$ for some k. From this we derive the fact that $c_{abs} \geq t$ if and only if $C[i,j] > 0$ for some i and j. We can repeatedly halve the possible range $[\alpha, \beta]$ of c_{abs} by adjusting the threshold value of t through the binary search. The algorithm is summarized as follows.

Algorithm 7. *Algorithm by binary search*

```
$\alpha = 0$
$\beta = n$
while $\beta - \alpha > 0$
   $t = (\beta + \alpha)/2$
   $b[i,j] = 1$ if $d^*[i,j] > t$, 0 otherwise for $i, j = 1, ..., n$
   Compute $C = B \times B$
   if $c[i,j] > 0$ for some $i$ and $j$
      $\alpha = \alpha + (\beta - \alpha)/2$
   else $\beta = \beta - (\beta - \alpha)/2$
end
$c_{abs} = \alpha$
```

Obviously the iteration in the while loop is done $O(\log n)$ times. Thus the total time excluding APSP becomes $O(B(n) \log n)$, where $B(n)$ is the time for multiplying (n, n) Boolean matrices. Let $M(n)$ be the time for the APSP with unit edge costs. Then the total time including APSP becomes $O(M(n) + B(n) \log n)$, meaning that the APSP is the bottle neck, as the best known complexity for APSP with unit edge costs is $O(n^{2.575})$ and that of $B(n)$ is $O(n^{\omega})$ with $\omega = 2.376$. Thus the APSP is the bottleneck with $O(n^{2.575})$.

When edge costs are in the range of $[0, \gamma]$ for a positive integer γ, we can initialize $\beta = \gamma n$ in the above algorithm, resulting in the time of $O(B(n)(\log n + \log \gamma))$, excluding the APSP. The best time for the APSP for general γ is $O(\gamma^{1/(4-\omega)} n^{2+1/(4-\omega)})$, which is the APSP bottleneck in this case. See [19] for the APSP complexities.

9 Concluding Remarks

We showed an asymptotic improvement on the time complexity of the two versions of 2-center problems; absolute 2-center and average 2-center, both of which take $O(n^3 \log \log n / \log n)$ time. As there are some algorithms for the APSP problem whose complexity is better than $O(n^3 \log \log n / \log n)$ [4], [10], etc., there may be some room for further improvement of asymptotic complexity for our problems.

If edge costs are small non-negative integers, the complexity for APSP becomes deeply sub-cubic. Once the APSP problem is solved using those sub-cubic time algorithms, the values in matrix D^*, the all-pairs shortest distance matrix, are no-longer small integers; they can be $O(n)$ or more, even if edge costs are all one. To overcome this increase of the values of matrix elements, we used the binary search idea for the absolute problem. It is not known whether we can use the same idea for the average problem.

The next step of research would be to extend the algorithm to the k-center problem. For a heuristic approach we propose to use an efficient algorithm for the 2-center algorithm repeatedly for the given graph, starting from the original graph. Then divide the set of vertices into two parts; one is the set of vertices closer to one center, and the other closer to the other center. Let G_1 and G_2 be the two sub-graphs induced from these two sets. If $c_{abs}(c_{ave})$ for G_1 is greater than that for G_2, then we solve the 2-center problem for G_1, otherwise for G_2. We can continue this process of dividing the set of vertices with the largest value of $c_{abs}(c_{ave})$ $k - 1$ times for $k \geq 2$. The computing time by this approach is $O(kT(n))$ where $T(n)$ is the time for the 2-center problem, but optimality cannot be guaranteed. Thus it is our concern how close to optimal the solution is. By experiments we observe that in case of the absolute problem this approximation algorithm achieves 1.2 times the optimal value for randomly generated complete graph with $k = 4$ and $n = 64$. For practical applications, graphs are more constrained, such as planar, satisfying Euclidean distance rule, hierarchical structure, etc. It remains to be seen whether these constraints serve for better approximation ratio by this heuristic.

References

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley, Reading (1974)
2. Alon, N., Galil, Margalit, O.: On the Exponent of the All Pairs Shortest 2Path Problem. Jour. Comp. Sys. Sci. 54(2), 255–262 (1999)
3. Ben-Moshe, B., Bhattacharya, B., Shi, Q.S.: Efficient Algorithms for the Weighted 2-Center Problem in a Cactus Graph. In: Deng, X., Du, D.-Z. (eds.) ISAAC 2005. LNCS, vol. 3827, pp. 693–703. Springer, Heidelberg (2005)
4. Chan, T.: More algorithms for all pairs shortest paths. In: STOC 2007, pp. 590–598 (2007)
5. Dobosiewicz, A.: more efficient algorithm for min-plus multiplication. Inter. J. Comput. Math. 32, 49–60 (1990)
6. Frederickson, G.N.: Parametric Search and Locating Supply Centers in Trees. In: Mosses, P.D., Schwartzbach, M.I., Nielsen, M. (eds.) CAAP 1995, FASE 1995, and TAPSOFT 1995. LNCS, vol. 915, pp. 299–319. Springer, Heidelberg (1995)
7. Fredman, M.: New bounds on the complexity of the shortest path problem. SIAM Jour. Computing 5, 83–89 (1976)
8. Garey, M.R., Johnson, D.S.: Computers and Intractability. W.H. Freeman, New York (1979)
9. Han, Y.: Improved algorithms for all pairs shortest paths. Info. Proc. Lett. 91, 245–250 (2004)
10. Han, Y.: An $O(n^3(\log\log n/\log n)^{5/4})$ Time Algorithm for All Pairs Shortest Path. Algorithmica 51(4), 428–434 (2008)
11. Kariv, O., Hakimi, S.L.: An Algorithmic Approach to Network Location Problems. I: The p-Centers. SIAM Jour. Appl. Math. 37(3), 513–538 (1979)
12. Kariv, O., Hakimi, S.L.: An Algorithmic Approach to Network Location Problems. II: The p-Medians. SIAM Jour. Appl. Math. 37(3), 539–560 (1979)
13. Takaoka, T.: A New upper bound on the complexity of the all pairs shortest path problem. Info. Proc. Lett. 43, 195–199 (1992)
14. Takaoka, T.: Subcubic algorithms for the all pairs shortest path problem. Algorithmica 20, 309–318 (1998)
15. Takaoka, T.: A faster algorithm for the all-pairs shortest path problem and its application. In: Chwa, K.-Y., Munro, J.I.J. (eds.) COCOON 2004. LNCS, vol. 3106, pp. 278–289. Springer, Heidelberg (2004)
16. Takaoka, T.: An $O(n^3\log\log n/\log n)$ time algorithm for the all pairs shortest path problem. Info. Proc. Lett. 96, 154–161 (2005)
17. Zwick, U.: All pairs shortest paths in weighted directed graphs - exact and almost exact algorithms. In: 39th FOCS, pp. 310–319 (1998)
18. Zwick, U.: A slightly improved sub-cubic algorithm for the all pairs shortest paths problem with real edge lengths. In: Fleischer, R., Trippen, G. (eds.) ISAAC 2004. LNCS, vol. 3341, pp. 921–932. Springer, Heidelberg (2004)
19. Zwick, U.: All pairs shortest paths using bridging sets and rectangular matrix multiplication. Jour. ACM 49(3), 289–317 (2002)

Appendix. Let us rename A_{ik} and B_{kj} in (4) by A and B. Let the difference lists, $\{a_{ir} - a_{is} | i = 1, ..., m\}$ and $\{b_{sj} - b_{rj} | j = 1, ..., m\}$, be sorted. Let $H_{rs}(i)$ and $L_{rs}(j)$ be the rank of $a_{ir} - a_{is}$ and $b_{sj} - b_{rj}$ in the list obtained by merging the above two sorted lists. Observe that

$$a_{ir} + b_{rj} \leq a_{is} + b_{sj} \Longleftrightarrow a_{ir} - a_{is} \leq b_{sj} - b_{rj} \Longleftrightarrow H_{rs}(i) \leq L_{rs}(j)$$

A sketch of the algorithm is to sort the difference lists in advance, and use the ranks of the data in a packed form in a single computer word. To determine the index k that gives the minimum to each element of the $(min, +)$-product for small matrices, we use a pre-computed table in $O(1)$ time, since the relative order of the above mentioned ranks can determine the index. An important observation is that sorting is done on data from A and B separately to minimize the time spent when A and B interact to produce the product.

In [16], it is shown that $T(m) = O(m^2(m \log m)^{1/2})$ with $m = O(\log n/(\log \log n)^3)$. Thus the time becomes $O(n^3(\log m/m)^{1/2})$. By the method of table look-up, it is shown in [16] that we can make the table in $O(n)$ time with $m = O(\log^2 n/ \log \log n)$, resulting in the total time of $O(n^3 \log \log n/ \log n)$ for the $(min, +)$-multiplication.

A Genetic Algorithm for Integration Planning of Assembly Operation and Machine Layout

C.J. Chiang[1], Z.H. Che[1], Tzu-An Chiang[2], and Zhen-Guo Che[3]

[1] Department of Industrial Engineering and Management,
National Taipei University of Technology,
1, Sec. 3, Chung-hsiao E. Rd., Taipei, Taiwan 106, ROC
wind112233@yahoo.com.tw
zhche@ntut.edu.tw
[2] Department of Commerce Automation and Management,
National Pingtung Institute of Commerce,
51 Min-Sheng E. Road, Pingtung, Taiwan 900, ROC
phdchiang@gmail.com
[3] Institute of Information Management,
National Chiao Tung University,
1001, University Rd., Hsinchu, Taiwan 300, ROC
bcc102a@hotmail.com

Abstract. This study is intended to examine the issue regarding the integration of assembly operations and machine layouts. For this issue, we consider a scenario of multiple orders and therefore build up an optimized mathematical model of synchronous planning. In the study, we develop an optimization mathematical model that uses a genetic algorithm (GA) for finding solutions and identify the most proper parameter value that best fit the GA by means of the experimental design. Ultimately, we use a case for a methodological application and the result shows that the GA may effectively integrate assembly operations and machine layouts for finding solutions.

Keywords: assembly operation, machine layout, genetic algorithm.

1 Introduction

In an era where global competition is drastically intense, manufacturers inevitably have to achieve high flexibility and optimized management for the production line to enhance product competitive edge. According to Beach et al. (2000), machine flexibility refers to the capability of a machine to swiftly perform various operations in the process [2]. If a machine is able to deal with multiple works, it indicates that the machine flexibility is relatively high. Routing flexibility refers to the possibility of a job operation to be processed by several resources. Manufacturing flexibility depends on machine flexibility and routing flexibility. Chang et al. (2003) propose two categories of manufacturing flexibility: (1) external-oriented manufacturing flexibility that directly reflects environmental changes in market; (2) internal-oriented manufacturing flexibility critical for market competitiveness [4].

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 533–543, 2010.

Focused on internal-oriented manufacturing flexibility, this study uses well-organized assembly planning for sufficient use of machines to improve production efficiency. In addition, Tompkins et al. (1996) argues that a proper layout may cut 10%-30% of production costs [8]. Chiang and Kouvelis (1996) suggest that there are 30%-70% of total manufacturing costs classified as material handling and layout [5]. Sarker et al. (2008) mention that a machine layout is a critical point for improving productivity in any manufacturing system [7]. A machine layout has great impact on the process in subsequent operations and therefore becomes a key issue for production lines.

In previous studies, there have been many scholars involved in wide discussion about problems related to operation planning and machine layouts with acceptable study results obtained. Wahab (2005) developed the flexibility models of the machine flexibility and product mix flexibility response under dynamic manufacturing environments [9]. He considered the machine-operation efficiency in the machine flexibility model. The tooling requirements, the number of operations, and the efficiency of different machines between different products are introduced in the product mix flexibility model. The comparison results showed that the proposed models were more comprehensive. Chan et al. (2005) developed an assignment and scheduling model to solve machining flexibility problems [3]. They developed an approach based on genetic algorithm for that problems and the results suggested machining flexibility can improve the production performance. Balakrishnan et al. (2003) presented a heuristic approach with effective and user friendly for solving the facility layout problem [1]. It is an extension of the static plant layout problem with unequal–sized departments. They used simulated annealing and genetic algorithm to solve the unequal-sized layout problem, and the results observed that genetic algorithm was better. McKendall Jr. and Hakobyan (2010) researched in the dynamic facility layout problem that is a multi-period and unequal-sized department layout problem [6]. They developed the tabu search and boundary search heuristic to improve the solution, and the results showed the proposed has the quality solution for that problems.

However, these studies were simply targeted at a single problem generally, no synergy of synchronous planning was observed. For this reason, this study incorporates machine layouts into the problem of assembly operations and wishes to boost the overall efficiency of production lines by synchronous planning. To address the integrated planning problem, this study is designed to build an optimization mathematical model and use a GA for finding solutions. With the study design, it is hoped to find a quality planning result in the scenario of perplexity.

The rest of this paper is organized as follows. Section 2 is the assumptions and mathematical foundation for the proposed integration problems. Section 3 details the flow of GA. Section 4 presents the case study to illustrate the experimental design and execution using parameters of GA for obtaining the optimal strategies. Results thus obtained the proposed GA. Section 5 contains the conclusions and suggestions for future studies.

2 Assumptions and Model

In this section, we will build an optimization mathematical model for integration of problems related to assembly operation planning and machine layouts. Assumptions are as follows:

1) During the production period, the processing time and efficiency are fixed and given.
2) The operations required for each order and the sequence are given.
3) The types and the quantity of machines in the factory are given.

Below are symbols used for developing the optimization mathematical model:

l	Order index, l=1,2,...,L
L	Total orders
j	Operational index, j=1,2,...,J
J	Total operations
t	Machine type index, t=1,2,…,T
T	Total machine types
k	The kth machine, k=1,2,...,K
K	Total machines
s,u	Location index, s=1,2,...,S, u=1,2,...,S
S	Total locations
O_{lj}	The jth operation of the lth order
M_{tk}	The kth machine of the tth type
ST_{lj}	Standard operating time of Operation O_{lj}
EM_{ljtk}	Handling efficiency of Operation O_{lj} at Machine M_{tk}
MT_{ljtk}	Actual handling time of Operation O_{lj} at Machine M_{tk}
TT_{ljtk}	Delivery time from Operation at O_{lj} at Machine M_{tk} to Operation O_{lr} at Machine M_{tk}
T_{su}	Delivery time from Location s to Location u
N_{il}	Total machines for the lth order
CT_l	Cycle time for the lth order
IT_l	Idle time for the lth order
TT_l	Delivery time for the lth order
SO_{tk}	Handling operation available at Machine M_{tk}
SM_{lj}	Machine available for handling Operation O_{lj}
X_{ljtk}	$\begin{cases} 1 & \text{Operation } O_{lj} \text{ assigned to Machine } M_{tk} \\ 0 & \text{Otherwise} \end{cases}$
Y_{ltks}	$\begin{cases} 1 & \text{Operation } O_{li} \text{ assigned to Machine } M_{tk}\text{, distributed to Location } s \\ 0 & \text{Otherwise} \end{cases}$

In the optimization model, the least sum of the cycle time and idle time of multiple orders will be taken into account. The complete mathematical model is provided below:

Objective function

$$Min\ Z = \sum_{l=1}^{L} [CT_l + IT_l] \tag{1}$$

$$CT_l = Max\ \{MT_{ljtk} + TT_{ljtk} \mid j=1,2,\ldots,J,\ t=1,2,\ldots,T,\ k=1,2,\ldots K\}\quad \forall\, l \tag{2}$$

$$MT_{ljtk} = \frac{ST_{lj}}{EM_{ljtk}} X_{ljtk}\quad O_{lj} \in SO_{tk} \tag{3}$$

$$TT_{ljtk} = [Y_{ljtks} Y_{l(j+1)tku} T_{su} \mid s=1,2,\ldots,S,\ u=1,2,\ldots,U]\quad \forall\, l, \forall\, j, \forall\, t, \forall\, k \tag{4}$$

$$IT_l = [CT_l N_l - \sum_{jtk} MT_{ljtk}]\quad \forall\, l \tag{5}$$

s.t.

$$\sum_{tk} X_{ljtk} = 0\quad \forall\, l, \forall\, j,\ M_{tk} \notin SM_{lj} \tag{6}$$

$$\sum_{lj} X_{ljtk} \geqq 1\quad \forall\, t, \forall\, k \tag{7}$$

$$\sum_{tk} X_{ljtk} = 1\quad \forall\, l, \forall\, j \tag{8}$$

$$\sum_{\mathrm{s}} Y_{ljtks} = 1\quad \forall\, l, \forall\, j, \forall\, t, \forall\, k \tag{9}$$

Objective function (1) is the least sum of the cycle time and machine idle time. Eq. (2) indicates the maximum sum of the processing time and handling time, defined as the cycle time. Eq. (3) indicates the actual handling time of operation at the machine. Eqs. (4) and (5) are the calculations of the handling time and idle time respectively. Eq. (6) indicates that an operation can only be distributed to a machine where the operation is handled. Eq. (7) indicates that there is at least an operation at a machine. Eq. (8) indicates that each operation is only assigned to a machine. Eq. (9) indicates that each machine is only assigned to a location.

3 Genetic Algorithm for Assembly Operation and Machine Layout Problems

To address the problems related to integrate operation planning and machine layout, this study uses the genetic algorithm for solutions. The step-by-step procedure for the algorithm is shown in Figure 1 and expressed as follows:

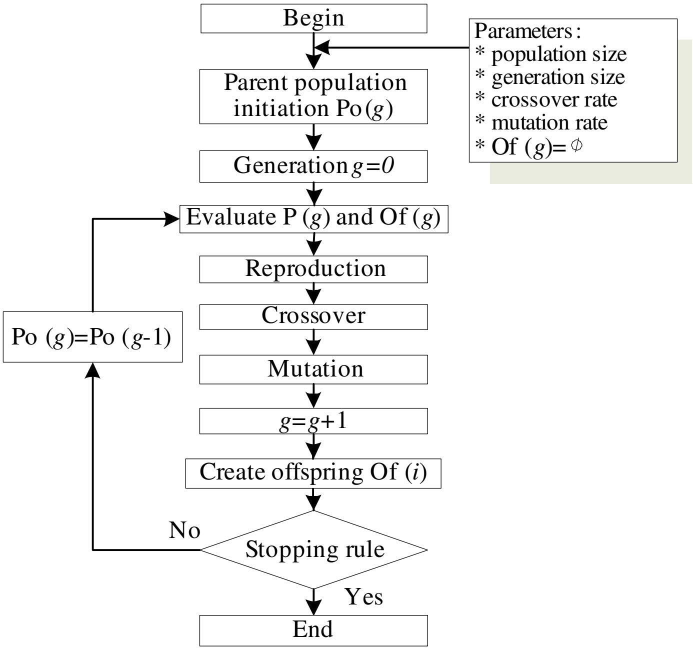


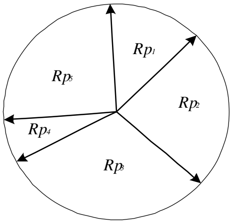
Fig. 1. Procedure of GA

Step 1: Encoding: a real encoding for the chromosome of the operation planning and machine layout.

Step 2: Generating initial population: The initial population is generated randomly provided it satisfies various constraints (Eqs. (6)-(9)).

Step 3: Calculating fitness function: Substitute the value of each individual into the optimized model. Inverse the objective function Z obtained from this model to be the fitness function of each individual.

Step 4: Reproduction: We use the Roulette Wheel rule which is the most popular in general genetic algorithms for duplication. Based on the calculation in the previous step, the percentage of each individual's fitness function to the total fitness function is the percentage of being selected; the higher the fitness function value, the easier the individual is likely to be selected. The concept of Roulette Wheel rule is shown in Figure 2.



$$Rp_i = \frac{Fv_i}{\sum_{i=1}^{n} Fv_i}$$

n: total number of chromosome

Population	Fitness Value	Reproduction Probability
Chromosome 1	Fv_1	Rp_1
Chromosome 2	Fv_2	Rp_2
Chromosome 3	Fv_3	Rp_3
Chromosome 4	Fv_4	Rp_4
Chromosome 5	Fv_5	Rp_5

Fig. 2. The Concept of Roulette Wheel Rule

Step 5: Crossover: We use one-point crossover in the GA. First, create a point of tangency on the chromosome randomly and then swap the genetic codes derived from the point of tangency. The concept of crossover is shown in Figure 3.

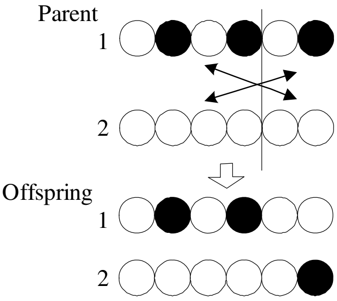


Fig. 3. The Concept of Crossover

Step 6: Mutation: A mutation site is generated randomly where the genetic code is randomly changed within a reasonable range. The concept of crossover is shown in Figure 4.

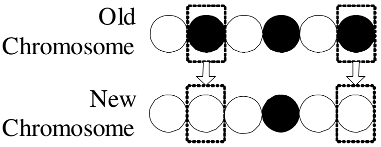


Fig. 4. The Concept of Mutation

Step 7: Generating new population: Individuals after evolution will become the new population, including the new operation planning and new machine layout.

Step 8: Whether termination condition is obtained: The termination condition used is the number of generations executed. The number of generations should be input before operation. It will stop as soon as the number of evolution times reaches the number of generations.

Step 9: Optimal strategy output: From the execution result, the optimal operation arrangement and machine layout will be derived.

4 Application of Cases and Analysis

4.1 Case Description

This study examines a case of Company A with assembly operations for two orders and the sizes of the factory locations. Company A has eight machines which are categorized into two types. Type 1 has four machines: *m1, m2*, *m3* and *m4*. Type 2 also has four: *m5, m6, m7* and *m8*. The layout of the factory space is similar to multi-line square, not rectangular, as shown in Figure 5. The eight locations are denoted as *L1-L8*.

L1	*L4*	*L6*
L2	*L5*	*L7*
L3		*L8*

Fig. 5. Factory Layout Location

Table 1. Machine Operating Efficiency for Order1

Type of machines	Machine	Order 1-efficiency of operation (%)							
		p1	*p2*	*p3*	*p4*	*p5*	*p6*	*p7*	*p8*
Type 1	*m1*	90	90	90	90	95	95	95	95
	m2	90	90	90	90	95	95	95	95
	m3	80	85	70	70	80	80	75	75
	m4	85	85	75	75	70	70	80	80
Type 2	*m5*	65	65	70	70	60	60	70	70
	m6	65	65	75	75	65	65	70	65
	m7	70	70	70	70	65	75	80	80
	m8	70	70	75	75	70	70	65	65
Standard operation time (Sec)		600	450	320	430	620	400	350	750

There are eight operations, *p1-p8*, for Order 1 and ten operations, *p1-p10,* for Order 2. The efficiency of each operation depends on the function of each type of machine. When the operating efficiency is 0%, it suggests that the operation is not available for the machine. Operations are assigned to machines according to their status, and then the machines are arranged at the factory locations. We calculate the objective value and complete the machine layout and operation assignment. Operation data are summarized in Tables 1-3.

Table 2. Machine Operating Efficiency for Order 2

Type of machines	Machine	Order 2-efficiency of operation (%)									
		p1	*p2*	*p3*	*p4*	*p5*	*p6*	*p7*	*p8*	*p9*	*p10*
Type 1	*m1*	80	85	70	70	60	60	75	75	0	0
	m2	85	70	80	80	85	85	80	80	75	75
	m3	90	95	90	90	90	90	95	90	95	95
	m4	90	80	80	90	90	90	90	95	90	90
Type 2	*m5*	65	65	70	70	75	75	75	80	70	70
	m6	80	80	75	75	75	65	65	65	75	75
	m7	70	70	70	70	70	70	75	75	80	80
	m8	70	70	75	75	60	60	60	60	80	80
Standard operation time (Sec)		630	420	500	350	400	550	620	800	400	320

Table 3. Location Material Handling Time Matrix

Code of factory locations	Material Handling Time (Sec)							
	L1	*L2*	*L3*	*L4*	*L5*	*L6*	*L7*	*L8*
L1	0	30	60	30	45	60	78	90
L2	30	0	30	45	30	78	60	78
L3	60	30	0	78	45	90	78	60
L4	30	45	78	0	30	30	45	78
L5	45	30	45	30	0	45	30	45
L6	60	78	90	30	45	0	30	60
L7	78	60	78	45	30	30	0	30
L8	90	78	60	78	45	60	30	0

4.2 GA Parameter Setting

GA parameters include population size, number of generations, crossover rate and mutation rate. To achieve the optimal efficiency for the decision system, it is necessary to conduct experimental designs for each parameter. For experimental issues, the experiment is designed to obtain the optimal parameter combination, so that we may provide a quick and effective solution for the GA proposed in this study. The proposed GA is coded with VB2005 programming language, and run on Pentium 4 CPU 3.2 GHz with 1 GB RAM.

First, the experiment of the crossover rate and mutation rate is conducted. Experimental parameters for the crossover rate are 0.5, 0.6, and 0.7, and experimental parameters for the mutation rate are 0.1, 0.2, and 0.3. The population size is 200. The number of generations is set to 500. The experimental results are shown in Table 4. Different crossover rates and mutation rates may lead to different solutions. Of various solutions, the optimal values were found to be 0.7 for the crossover rate and 0.3 for the mutation rate.

Table 4. Experiment Result of Crossover Rate and Mutation Rate

Mutation	Crossover rate		
rate	0.5	0.6	0.7
0.1	4021.623	3952.902	4011.983
0.2	3913.590	3936.112	4021.178
0.3	3943.790	3923.413	3886.070

The experimental design of the population size follows. Experimental parameters for the population size are 100, 200 and 300. Parameters for the crossover rate and mutation rate are the same as those obtained from the last experiment: 0.7 and 0.3, and the number of generations is 500. According to the result, the decrease in solution quality is observed when the population size is reduced from 200 to 100 and the increase is observed when the size is enlarged from 200 to 300. Thus, we have the highest solution quality as the population size is 300, as shown in Table 5.

Table 5. Experiment Result of Population Size

Population number	100	200	300
The average value of objective function	4002.926	3886.07	3873.869

In the last part, we conduct the experimental design of the number of generations. Experimental parameters for the number of generations are 400, 500, and 600. The crossover rate, mutation rate and the population size are the same as those obtained in the preceding two experiments. According to the result shown in Table 6, solution quality does not improve as the number of generations increases from 500 to 600 or decreases from 500 to 400; so we may achieve the best benefit when the number of generations is set to 500.

Table 6. Experiment Result of Number of Generations

Generation number	400	500	600
The average value of objective function	3887.175	3873.869	3878.666

From these experimental results, we may conclude the optimal genetic parameter settings for the decision model: population size: 300; number of generations: 500; crossover rate: 0.7; and mutation rate: 0.3. The optimal mean of the objective function value is 3873.869.

4.3 Results

The optimal parameter setting is obtained from the experimental design. The GA is implemented with the previous data and parameters to obtain the operation planning and machine layout. Figure 6 shows the best operation arrangement and overall machine layout, in which Figure 6(a-1) and Figure 6(a-2) represent for Order 1 and Order 2 respectively. As shown in Figure 6, Machine m4 is assigned to Location *L1*, processing Operation *p1* for Order 1 and Operation *p8* for Order 2. The arrows indicate the directions of the operation flow. According to the results of the illustrative example, the operation starts from *p1* to *p8* for Order 1. *p1* is assigned to *m4* in *L1*, *p2* is assigned to *m6* in *L3*, and so on. The operation starts from *p1* to *p10* for Order 2. *p1* is assigned to *m7* in *L7*, *p2* is assigned to *m3* in *L5*, and so on. The optimal objective function value in the experiment is 3835.21, where the cycle times for Order 1 and Order 2 are 789.47 and 976.67, and the idle times are 1198.94 and 870.13 respectively.

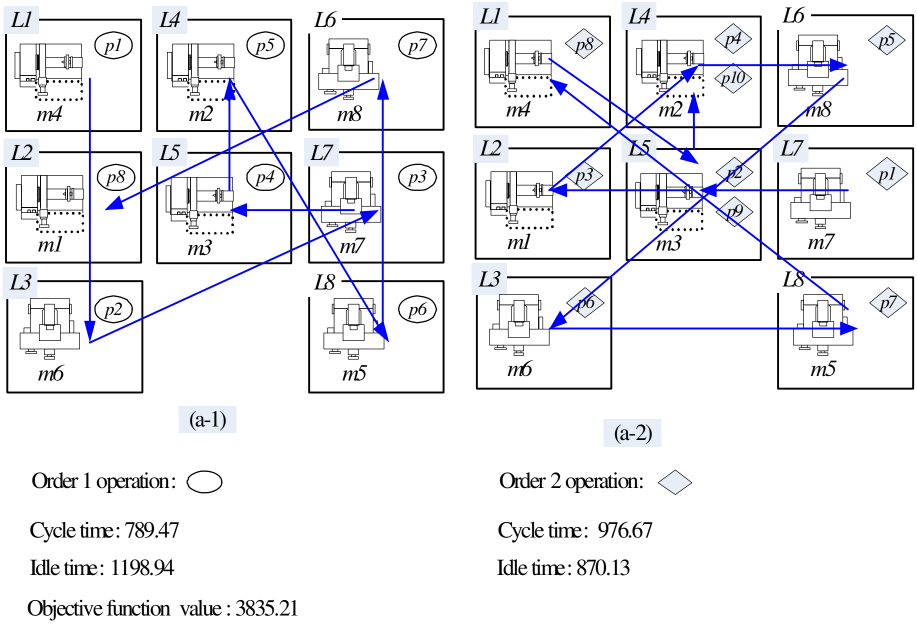


Fig. 6. Optimal Operation Planning and Machine Layout

5 Conclusions and Suggestions

This study mainly examines the integrated planning problem, which synchronizes with assembly operation planning and machine layout planning. For the planning, we expect to achieve the least sum of the cycle time and idle time when considering a scenario of multiple orders. In this study, an optimized mathematical model was built, systemically describing the problem for solving. Moreover, this study applies the GA to find solutions for the sophisticated mathematical model. Ultimately, we introduce the developed mathematical model and GA to cases. According to the results, the GA can be used to effectively solve the integrated problem.

Throughout the study, we have also found some directions that deserve being further researched:

1) Consider other related production information, e.g. failure rate of machine, capacity, and workable space.
2) Develop an optimization mathematical model with different weight for each objective.

References

1. Balakrishnan, J., Cheng, C.H., Wong, K.F.: FACOPT: A User Friendly FACility Layout OPTimization System. Comput. Oper. Res. 30(11), 1625–1641 (2003)
2. Beach, R., Muhlemann, A.P., Price, D.H.R., Paterson, A., Sharp, J.A.: A Review of Manufacturing Flexibility. Eur. J. Oper. Res. 122(1), 41–57 (2000)
3. Chan, F.T.S., Wong, T.C., Chan, L.Y.: A genetic algorithm-based approach to machine assignment problem. Int. J. Prod. Res. 43(12), 2451–2472 (2005)
4. Chang, S.C., Yang, C.L., Sheu, C.: Manufacturing Flexibility and Business Strategy: An Empirical Study of Small and Medium Sizes Firms. Int. J. Prod. Econ. 83(1), 13–26 (2003)
5. Chiang, W.C., Kouvelis, P.: Improved Tabu Search Heuristics for Heuristics Solving Facility Layout Problem. Int. J. Prod. Res. 34(9), 2585–2965 (1996)
6. McKendall Jr., A.R., Hakobyan, A.: Heuristics for the Dynamic Facility Layout Problem with Unequal-area Departments. Eur. J. Oper. Res. 201(1), 17–182 (2010)
7. Sarker, R., Ray, T., Da Fonseca, J.B.: An Evolutionary Algorithm for Machine Layout and Job Assignment Problems. 2007 IEEE Congress on Evolutionary Computation, CEC 2007 art. no. (4424991), 3991–3997 (2008)
8. Tompkins, J.A., White, J.A., Bozer, Y.A., Frazelle, E.H., Tanchoco, J.M., Trevino, J.: Facilities. Wiley, New York (1996)
9. Wahab, M.I.M.: Measuring machine and product mix flexibilities of a manufacturing system. Int. J. Prod. Res. 43(18), 3773–3786 (2005)

Diagnosis of Learning Styles Based on Active/Reflective Dimension of Felder and Silverman's Learning Style Model in a Learning Management System

Ömer Şimşek, Nilüfer Atman, Mustafa Murat İnceoğlu, and Yüksel Deniz Arikan

Ege University, Department of Computer and Instructional Technologies
35100 Bornova, Izmir, Turkey
omarsimsek@gmail.com, nilatman@hotmail.com,
mustafa.inceoglu@ege.edu.tr, deniz.arikan@ege.edu.tr

Abstract. Learner centered education is important both in point of face to face and Web based learning. Due to this importance, diagnosis of learning styles of students in web based or web enhanced educational settings is important as well. This paper presents prediction of learning styles by means of monitoring learner interface interactions. A mathematics course executed on a learning management system (Moodle) was monitored and learning styles of the learners were analyzed in point of active/reflective dimension of Felder and Silverman Learning Styles Model. The data from learner actions were analyzed through literature based automatic student modeling. The results from Index of Learning Styles and predicted learning styles were compared. For active/reflective dimension 79.6% precision was achieved.

Keywords: Learning styles, Felder and Silverman's Index of Learning Styles, Moodle, Web-enhanced learning.

1 Introduction

Individuals have different backgrounds, motivation and preferences in their own learning processes. Web-based systems that ignore these differences have difficulty in meeting learners' needs effectively [1]. Therefore, when designing instructional material, it is important to accommodate elements that reflect individual differences in learning. One of these elements is learning styles [2]. Learning styles of learners must be determined for adapting instructional material that best suits students learning styles. There are many ways of detecting learners learning style and mostly it is performed by questionnaires or scales. However, using questionnaires for determining learning styles has some disadvantages such as not all the students are motivated to fill out questionnaire. As a result, mismatches between real behavior and questionnaire answers could exist. To overcome these problems, instead of allocating time to filling out questionnaires, student models could be constructed. An alternative approach to collect the information pertinent to a student model is to track the student's behavior and responses and then make inferences about general domain competence, cognitive traits, and learning styles. The challenge of this approach is to identify and

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 544–555, 2010.

collect sufficient information to make reliable and useful inferences [3]. The aim of this paper is to automatically detect learning styles of learners through analyzing their behaviors in an LMS course.

2 Related Work

Detecting automatically learning style requires reviewing two main contexts such as learning styles and adaptive systems.

2.1 Learning Styles

There are many models about learning styles in literature such as Kolb [4], Dunn & Dunn [5], Honey & Mumford [6], Myers-Briggs [7]. This study is based on Felder and Silverman's Learning Styles Model because of its applicability to e-learning and compatibility to the principles of interactive learning systems design [8].

Students learn in many ways — by seeing and hearing; reflecting and acting; reasoning logically and intuitively; memorizing and visualizing and drawing analogies and building mathematical models; steadily and in fits and starts [9]. The ways in which an individual characteristically acquires, retains, and retrieves information are collectively termed the individual's *learning style* [10]. In 1988 R. Felder and L. Silverman proposed a learning style model that classifies five dimensions of learning styles.

Table 1. Dimensions of Felder and Silverman Learning Style Model

Dimensions of Felder and Silverman Learning Style Model	
Sensory	Perception
Intuitive	
Visual	Input
Verbal	
Active	Process
Reflective	
Sequential	Understanding
Global	

Sensory/intuitive dimension specified the type of the information that students prefer to perceive. The dimension of visual/auditory (lately amended as visual/verbal) is refer to trough which sensory channel is external information most effectively perceived; active/reflective represent how does the student process the information and finally sequential/global shows how does the student progress towards understanding [9].Lately, inductive/deductive was excluded from the model.

- sensing learners (concrete, practical, oriented toward facts and procedures) or intuitive learners (conceptual, innovative, oriented toward theories and meanings);

- visual learners (prefer visual representations of presented material--pictures, diagrams, flow charts) or verbal learners (prefer written and spoken explanations);
- active learners (learn by trying things out, working with others) or reflective learners (learn by thinking things through, working alone);
- sequential learners (linear, orderly, learn in small incremental steps) or global learners (holistic, systems thinkers, learn in large leaps) [11].

Although there are four dimensions in this model; 'process' dimension is investigated because this dimension's characteristics are supported in proposed LMS course. Due to having various collaborative and individual web activities, the study focused on active/reflective dimension of Felder and Silverman learning style model. The promotion of individual activities such as watching videos, viewing pdf documents, answering questionnaires, attempting quizzes and collaborative activities such as forum discussions are the reasons of choosing this dimension. Also, In Garcia et al. [12], the study results showed the low prediction achievement for active/reflective dimension because of the lack use of communication tools such as forums, chat, mail, etc. In this study, it is aimed at giving more detailed data for active/reflective dimension.

In process dimension, individuals prefer to process in two ways *actively*— through engagement in physical activity or discussion, or *reflectively*— through introspection [9]. Active learners learn well in situations that enable them to do something physical and reflective learners learn well in situations that provide them with opportunities to think about the information being presented [10]. Active learners work well in groups; reflective learners work better by themselves or with at most one other person. Active learners tend to be experimentalists; reflective learners tend to be theoreticians [9].

The Index of Learning Styles [ILS] was created in 1991 by Richard M. Felder, and Barbara A. Soloman, then it was installed on the World Wide Web in 1996. ILS is a 44 item questionnaire and there are eleven questions per four dimensions [13]. The ILS scales are bipolar, with mutually exclusive answers to items, i.e. either (a) or (b). Because there is an odd number of items on each scale, if items are scored as +1 and –1, respectively, the total score on a scale from –11 to +11 shows an emerging preference for the given modality [14]. If the score on a scale is 1 between 3, it indicates there is a balanced preference on the two dimensions of that scale. If the score on a scale is 5-7, it could be said there is a moderate preference for one dimension of the scale and will learn more easily in a teaching environment which favors that dimension. If the score on a scale is 9-11, there is a strong preference for one dimension of the scale and may have real difficulty learning in an environment which does not support that preference [13].

2.2 Student Modeling

In literature, there are adaptive educational hypermedia systems considering learning styles such as CS383 [15], MANIC [16], MASPLANG [17], AHA! [18], TANGOW [19]. Adaptive hypermedia systems build a model of the goals, preferences and knowledge of each individual user, and use this model throughout the interaction with the user, in order to adapt to the needs of that user [20]. The student modeling module performs two main functions: first, initializes the student model when a new student logs on the system for the first time, second, updates the student model based on the student's interaction with the system [21].

The majority of adaptive systems focusing on learning styles are using a collaborative student modeling approach by asking students to fill out a questionnaire for detecting their learning styles. While collaborative student modeling requires students to explicitly provide some information about their preferences and needs, an automatic student modeling approach is based on the concept of looking at what students are really doing in a course and inferring their preferences and needs from their behavior and actions in the course [22].

One of the recent researches that automatically diagnosing learning styles, belongs to Cha et al. [23]. In their research, they detected learning styles with data-driven approach based on user interface behaviors. System was based on Felder and Silverman Learning Styles Model and four dimensions of model were investigated. The interface of the system was designed with Macromedia Flash and interface behaviors of 70 participants were monitored. First, the learning styles of learners were detected then interface was adapted respecting learning styles.

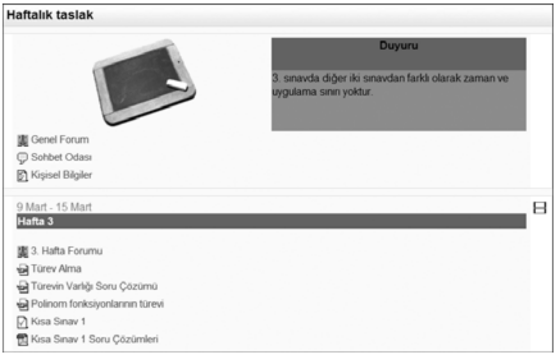
Another research, detecting learning styles with fully automatic student modeling belongs to Garcia et al. [12]. They used data-driven approach and evaluated Bayesian networks at detecting the learning styles of a student. The Bayesian network models different aspects of a student behavior while he/she works with the system. Then, it infers his/her learning styles according to the modeled behaviors [12]. Forum, chat, e-mail, reading materials (concrete or abstract), exam revisions, exercises, answer changes, exam results, etc., were investigated as behavior patterns. Also the system was based on Felder and Silverman's Learning Styles Model and three dimensions were investigated, namely; perception, process and understanding. 50 students were used to train Bayesian networks and system was tested by 27 students. Using the evaluation method which was developed by themselves, they obtained a precision of 77% in the perception dimension, 63% in the understanding dimension, and 58% in the processing dimension.

There is also an alternative way for fully automatic detection which was developed by Graf namely; literature based approach [22]. Differently from data-driven approach, literature based approach, is to use the behavior of students in order to get hints about their learning style preferences and then apply a simple rule-based method to calculate learning styles from the number of matching hints. This approach is similar to the method used for calculating learning styles in the ILS questionnaire and has the advantage to be generic and applicable for data gathered from any course due to the fact that Felder and Silverman Learning Style Model is developed for learning in general [22]. In the research, 75 students attended 'Object Oriented Modeling' course for seven weeks and student behaviors analyzed in a Learning Management System, with the help of literature based approach. Sensory/intuitive, active/reflective and sequential/global dimensions of Felder and Silverman's Learning Style Model were investigated. In this study, first behavior patterns which were frequently used in LMS were determined and thresholds for each patterns propounded with the help of experiments and reviewing literature. Data from student behaviors compared with thresholds and calculate matching hints for detecting learning styles. For evaluating how close the predicted values to ILS, the formula was developed by Garcia et al. [12] was used, and 79.33% prediction achievement was found for active/reflective, 77.33% for sensory/intuitive and 77.33% for sequential/global dimension.

Atman et al. [1] used literature-based approach for detecting learning styles in a web-based course. In the study, a web based education system proposed and each module is labeled for their corresponding learning style dimension. Each module is labeled such as Visual_Active, Visual_Reflective, Verbal_Active and Verbal_Reflective. This makes analyzing process faster and transportable to other dimensions. To evaluate system effectiveness, learners filled Index of Learning Styles Questionnaire at the beginning of the course. Scores of predicted learning styles and ILS scores are compared by using the formula developed by Garcia et al. [12]. 17 college students' behaviors are analyzed with literature based approach and results show that the prediction achievement is 83.15% for active/reflective dimension.

3 Automatic Detection of Learning Styles With Literature-Based Approach

In this study, behaviors of 27 freshmen enrolled in Mathematics course in department of computer education and instructional technologies were analyzed with literature-based approach and learning styles of learners automatically detected. The course which carried out in the spring semester of 2008-2009 was enhanced with web activities. The course took 14 weeks and during the course "Derivative" topic was chosen. Derivative is one of the important topics in calculus and students have difficulty in understand it. Therefore the face to face course was enhanced with an LMS including resources and activities online. In this course, interactions with forums (10), content including videos (37) and PDFs(6), questionnaires(5), quizzes(3) and user profile viewing were investigated.



Haftalık taslak

Duyuru

3. sınavda diğer iki sınavdan farklı olarak zaman ve uygulama sınırı yoktur.

Fig. 1. Screenshot of the course

The forum discussions were mostly about the topic "derivatives" and also included announcements about the course, social dialogs between students. All of the video contents had screen captured video lectures of derivative subject; without the lecturer's view but the voice with digital pen moves on whiteboard software. The PDF content included exam solutions, exam dates and exam scores.

Also forum postings and the time spent on forum discussions are considered. Forum and user profile actions are one of the frequently used tools in this course so data from these behaviors are monitored and analyzed. There are five questionnaires in this course such as personal information, views about derivative subject, video lesson assessment questionnaire, assessment of derivative subject and views about the teacher in the videos.

To evaluate student performance three short exams were used and time used for each exam and the total number of performed question for each exam are considered and determined as behaviors. The tendencies of students to view other student's profiles (classmates) were also investigated with regard to their clicking on a profile and the spent time on the activity.

3.1 Investigated Patterns

Six features are investigated in this study. Each pattern valued as '-' for reflective and '+' for active dimensions according to characteristics of the pattern. For determining the patterns, first the literature reviewed also new patterns such as patterns dealing with user profile is introduced in this study. In Content objects, both video and PDF, learners are expected to listen or read and learned by thinking through so the value of these patterns are marked as '-'. If one of the students spends more time or visit more than thresholds, it could be said, there is an evidence for reflective learning. While Content features have reflective properties, forum posting valued as '+' because active learners tend to retain and understand information best by doing, discussing or applying it or explaining it to others actively. On the other hand reflective learners learn well in situations that provide them with opportunities to think about the information being presented so it is expected to spend more time in forums for reflective learners. In this study, one of the frequently used features is user profile view, it is investigated as well. Visiting user profile is expected to increase, while the time spent on user profile view decrease for active learners. Performed quiz and questionnaire questions are considered as active characteristics and the time spent on quiz or questionnaires regarded as reflective. Table 2 shows the investigated patterns, patterns descriptions and relevant thresholds for each pattern.

After determining patterns for dimensions, relevant thresholds for each pattern must be defined to analyze learner behaviors systematically. The data that comes from the learner behavior is compared with these thresholds so it gives hints about learning styles of the learner. Thresholds are used as evaluation criteria for data coming from learner behaviors. Most of the thresholds values are determined with the help of reviewing literature. Thresholds for content objects are defined assuming that these objects are required to read in order to understand the topic; a value of 75% and 100% of the available content objects is recommended [22]. Visiting forums 7 to 14 per week and posting 1-10 is considered normal values [22], [24]. For user profile, no recommendations were given. It can be assumed 50% and 75% visit frequency of user profile view and 25% and 50% for time on user profile view. Thresholds for performed quiz and questionnaire questions are set to 25% and 75% based on the assumptions of Garcia et al. [12].

Table 2. Thresholds for Patterns

	Behavior Pattern	Pattern Description	Thresholds
Content Video	Content(video)_visit (-)	percentage of visited video content (based on the number of available content objects	75% - 100%
	Content(video)_stay (-)	percentage of time spent on video content (based on average time)	50% - 75%
Content PDF	Content(PDF)_visit (-)	percentage of visited PDF content objects (based on the number of available content objects	75%- 100%
	Content(PDF)_stay (-)	percentage of time spent on PDF content (based on average time)	50% - 75%
Forum	Forum_post (+)	Number of postings in the forum (per week)	1-10
	Forum_discussion_stay/visit (-)	Percentage of division of average staying time to visit frequency in a forum discussion.	50% - 75%
User Profile	Userview_visit (+)	Percentage of visited user profiles (based on average value)	50% - 75%
	Userview_stay (-)	Percentage of division time spent on user profiles to visit number (based on average value)	25% - 50%
Quiz	Quiz_visit (+)	percentage of performed quiz questions (based on the total amount of available questions)	25% - 75%
	Quiz_stay (-)	percentage of time spent on quiz (based on a predefined expected value)	50% - 75%
Questionnaire	Questionnaire_visit (+)	percentage of performed questionnaire questions (based on the total amount of available questions)	25% - 75%
	Questionnaire_stay (-)	percentage of time spent on questionnaire (based on a predefined expected value)	50% - 75%

It is recommended in Garcia et al. [12], the time spent on content objects, quiz and questionnaires are assumed as 50% and 75%.

When analyzing visiting frequencies in literature-based student modeling approach, total number of content is considered. The score of visiting frequency of learner proportioned to total number of the respecting content. This percentage compared with the thresholds. On the other hand, when analyzing the time spent on a feature, predefined values are used. In features like “Questionnaire” and “Quiz”, is determined, but some problems came up while determining expected values for user profile view. For this reason, average time was used as criteria in this study. Up and down values are excluded while averaging process to prevent affecting results negatively. For example, total video staying time of a learner is rated to a average value, which was obtained of values excluded top and low values, gives us a percentage. Then, this percentage is compared with thresholds.

The next step is to compare data coming from learner and thresholds and calculate hints to detect learning styles of the learner. For example if the percentage of number of total performed questions to all of the questions is between thresholds, this gives us there is balanced evidence. If the values are less than threshold, there is a weak preference for active and strong evidence for reflective learning. If the percentage of the learner is higher than threshold, it could be said there is a strong evidence for active and weak evidence for reflective learner.

3.2 Method of Evaluation

Evidences for active learning is marked as “3”, for balanced learning “2” is marked and finally “1” is pointed for reflective learning. The average of total hints ranged between 1 to 3 and these results are normalized 1 to 0; 1 for active and 0 for reflective learning. 0.25 and 0.75 were used as thresholds [22]. Later the results from Index of learning Styles are mapped into 3 – item scale and compared with the predicted values to see prediction precision.

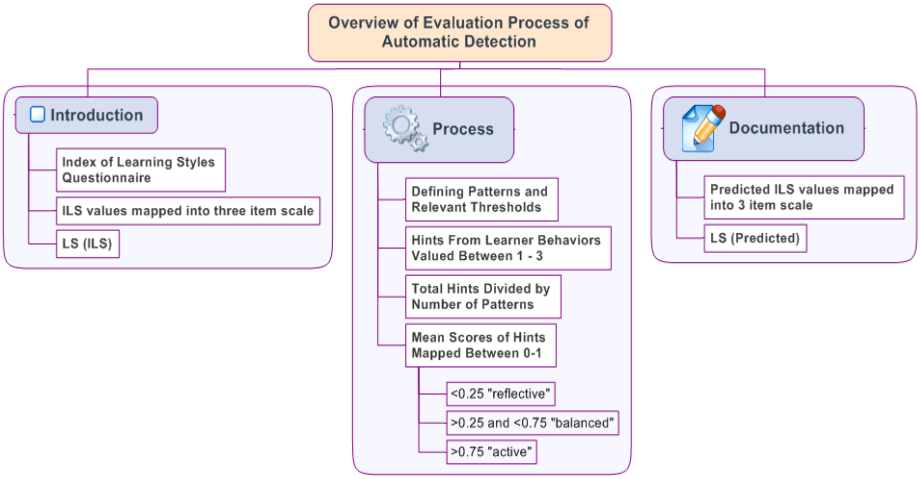

Overview of Evaluation Process of Automatic Detection
Introduction
Index of Learning Styles Questionnaire
ILS values mapped into three item scale
LS (ILS)
Process
Defining Patterns and Relevant Thresholds
Hints From Learner Behaviors Valued Between 1 - 3
Total Hints Divided by Number of Patterns
Mean Scores of Hints Mapped Between 0-1
<0.25 "reflective"
>0.25 and <0.75 "balanced"
>0.75 "active"
Documentation
Predicted ILS values mapped into 3 item scale
LS (Predicted)

Fig. 2. Overview of evaluation process of automatic detection [1]

Comparison of the values is done with the formula developed by Garcia et. al [12]. In this formula, if predicted learning styles and ILS values is equal, formula returns 1. If one the values is balanced and the other is a preferred learning style of the two poles of that dimension, function then returns 0.5. Finally if each of two values differs from each other, the function returns 0. This formula is performed for each student and divided to the number of learners.

3.3 Study Group

The study group consists of 19 male and 8 female students and 23 of them have their own computers. The prior knowledge of group members about the subject is 23 low, 4 moderate and 0 advanced. In acquiring information about their prior knowledge, a pretest was used. In this context, describing qualification of the group members regarding to students views may help understanding learning styles of students while discussing the results. Besides, 9 of the students have not taken any online enhanced course.

Table 3. Characteristics of the Study Group

Computer use skills	Internet use skills	Frequency of using Internet in a week	Frequency of controlling mail-box in a day	Frequency of sending mail in a day
Strongly insufficient(0)	Strongly insufficient (0)	Never (0)	Never (1)	Don't send mail (4)
Insufficient (2)	Insufficient (2)	0-1 hour (4)	Once (9)	1 mail (5)
Moderate (5)	Moderate (4)	1-3 hours (4)	Twice (2)	More than 2 mails (2)
Sufficient (10)	Sufficient (10)	3-5 hours (1)	No opinion (9)	No opinion (16)
Strongly sufficient (5)	Strongly sufficient (4)	Above 5 hours (11)		
No opinion (7)	No opinion (7)	No opinion (7)		

The table shows study groups' computer and Internet use skills and frequencies of using Internet in a week, controlling mailbox and sending mail in a day. The table shows us the members of group have sufficient computer use skills, sufficient Internet use skills, and spend time using Internet, mostly controlling mailbox once a day and sending very few mail in a day. Table 3 summarizes the characteristics of the study group. Although there are group members who have no opinion about the variables, it is assumed that skills and opportunities of students show normal distribution.

4 Results and Discussion

In this study, learner behaviors in an LMS course are analyzed through the help of literature-based approach. Felder and Silverman Learning Styles Model is chosen and active/reflective dimension of this model is investigated. 27 college students' behaviors analyzing results show that the precision is 79.63% for process dimension. Comparison of the ILS questionnaire result and diagnosed learning style is shown in Table 4. Bold italic parts, illustrate mismatches.

As seen in Table 4, most of the students have balanced (Neutral: NEU) learning styles so that literature-based approach detected learning styles balanced too. Table 5 shows the compare results of prediction achievement for this study and the other studies.

Table 4. Comparisons of ILS Results to Predicted Results

No	ILS Result	Predicted Result	No	ILS Result	Predicted Result
1	NEU	NEU	15	NEU	NEU
2	NEU	NEU	16	***REF***	***NEU***
3	***REF***	***NEU***	17	***REF***	***NEU***
4	NEU	NEU	18	NEU	NEU
5	NEU	NEU	19	***ACT***	***NEU***
6	NEU	NEU	20	NEU	NEU
7	NEU	NEU	21	NEU	NEU
8	NEU	NEU	22	***REF***	***NEU***
9	NEU	NEU	23	ACT	ACT
10	NEU	NEU	24	***REF***	***NEU***
11	***REF***	***NEU***	25	NEU	NEU
12	***ACT***	***NEU***	26	NEU	NEU
13	***REF***	***NEU***	27	***REF***	***NEU***
14	***ACT***	***NEU***			

Table 5. Comparison Results

Modeling Approach	**Web-based/LMS**	**Study**	**Participants**	**Precision for active/reflective**
Data-driven	Web-based	Garcia et al. [12]	27	62.50%
Literature-based	LMS	Graf [22]	75	79.33%
Literature-based	Web-based	Atman et al. [1]	17	83.13%
Literature-based	LMS	This study	27	79.63%

The table indicates that literature based approach has better precision prediction than the data-driven approach. In this study, the results have close values to other studies.

The number of members in the study group, the bandwidth opportunity, the prior knowledge about the subject may affect students' tendencies to stay in a online course activity. In addition to these limitations; lack of information about the students who have no opinion for computer (7) and Internet (7) use skills and frequency of using Internet in a week (7) are limitations of the study as well. Also, the 9 students who have not taken any online courses should be taken into consideration while predicting their learning styles with help of their stay time and click frequencies in a web activity. Because, a student who have never used an online course may spend more time to understand how a forum discussion works, how to attempt a quiz. So, the analysis results can assume active learner as balanced or as reflective.

Another limitation is the subject of the courses. The investigated course of this is study is Mathematics, in Atman et al.[1] learner behaviors of an English course was analyzed, while in Graf [22], "Object Oriented Modeling", and in Garcia et al. [12], "Artificial Intelligence" were chosen subject of the courses. Although there are many limitation variables for predicting learning style, the results show a high precision for active/reflective dimension.

5 Conclusion

Diagnosing learning styles of students automatically through analyzing learner behaviors in an LMS course is important from pedagogical aspect. The study based on Felder and Silverman Learning Styles Model and process dimension is investigated through the help of literature-based approach. In this way, the data of learning styles of learners can be provided to adaptive systems. Predicted scores and the ILS scores are compared.

While results demonstrated promising outcomes, also it is not expected to achieve 100% prediction with analyzing staying time and visiting frequency of students. Real life behaviors and web behaviors can differ in sometime. It is said, there could be differences with questionnaire items which are consist of real life behaviors and actions (i.e. clicking, staying time) and actions in a web environment. To cope with these problems, it is more suitable to compare learning styles with a psychometric tool which was developed to evaluate especially web actions of learners. With this type of measurement instrument could be presented, when a learner first time logged in to LMS. By this way, course administrators and course instructors can monitor learning styles of learners and this information can help them to plan and organize instructional activities more effectively. In addition to improving an online learning style scale, it is important to determine ones learning style while spending time and clicking on a web activity, it is important to know computer and Internet use skills and prior knowledge. These levels affects students' tendency to join Web activities, as well.

References

1. Atman, N., Inceoğlu, M.M., Aslan, B.G.: Learning Styles Diagnosis Based On Learner Behaviors in Web Based Learning. In: Gervasi, O., Taniar, D., Murgante, B., Laganà, A., Mun, Y., Gavrilova, M.L. (eds.) Computational Science and Its Applications – ICCSA 2009. LNCS, vol. 5593, pp. 900–909. Springer, Heidelberg (2009)
2. Bajraktarevic, N., Hall, W., Fullick, P.: Incorporating Learning Styles in Hypermedia Environment: Empirical Evaluation. In: de Bra, P., Davis, H.C., Kay, J., Schraefel, M. (eds.) Proceedings of the Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, pp. 41–52. Eindhoven University, Nottingham (2003)
3. Graf, S., Lin, T.: Kinshuk: The Relationship Between Learning Styles and Cognitive Traits. Computers in Human Behavior 24, 122–137 (2008)
4. Kolb, A.Y., Kolb, D.A.: The Kolb Learning Style Inventory - Version 3.1, Technical Specification. Hay Group, Boston (2005)

5. Dunn, R., Dunn, K., Price, G.E.: Learning Style Inventory, Lawrence, KS. Price Systems (1996)
6. Honey, P., Mumford, A.: The Learning Styles Helper‘s Guide. Peter Honey Publications Ltd., Maidenhead (2006)
7. Myers, I.B., McCaulley, M.H.: Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator. Consulting Psychologists Press, Palo Alto (1998)
8. Kuljis, J., Liu, F.: A Comparison of Learning Style Theories on the Suitability for Elearning. In: Hamza, M.H. (ed.) Proceedings of the Iasted Conference on Web Technologies, Applications, and Services, pp. 191–197. ACTA Press (2005)
9. Felder, R., Silverman, L.: Learning and Teaching Styles. Journal of Engineering Education 94(1), 674–681 (1988)
10. Felder, R., Henriques, E.R.: Learning and Teaching Styles In Foreign and Second Language Education. Foreign Language Annals 28(1), 21–31 (1995)
11. Felder, R.M.: Matters of Style. ASEE Prism 6(4), 18–23 (1996)
12. García, P., Amandi, A., Schiaffino, S., Campo, M.: Evaluating Bayesian Networks Precision for Detecting Students Learning Styles. Computers &Education 49(3), 794–808 (1995)
13. Felder, R.M., Soloman, B.A.: Index of Learning Styles Questionnaire (October 2009), `http://www.engr.ncsu.edu/learningstyles/ilsweb.html`
14. Zywno, M.S.: A Contribution to Validation Score Meaning for Felder-Soloman‘s Index of Learning Styles. In: Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition (2003)
15. Carver, C.A., Howard, R.A., Lane, W.D.: Addressing Different Learning Styles through Course Hypermedia. IEEE Transactions on Education 42(1), 33–38 (1999)
16. Stern, M.K., Steinberg, J., Lee, H.I., Padhye, J., Kurose, J.: Manic: Multimedia Asynchronous Networked Individualized Courseware. In: Proceedings of the World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications (Ed-Media/Ed-Telecom), Calgary, Canada, pp. 1002–1007 (1997)
17. Peña, C.-I., Marzo, J.-L., de la Rosa, J.-L.: Intelligent Agents in a Teaching and Learning Environment on the Web. In: Petrushin, V., Kommers, P., Kinshuk, G.I. (eds.) Proceedings of the International Conference on Advanced Learning Technologies, pp. 21–27. IEEE Learning Technology Task Force, Palmerston North (2002)
18. Stash, N., Cristea, A., de Bra, P.: Authoring of Learning Styles in Adaptive Hypermedia: Problems and Solutions. In: Proceedings of the International World Wide Web Conference, pp. 114–123. ACM Press, New York (2004)
19. Carro, R.M., Pulido, E., Rodriguez, P.: Tangow: A Model for Internet-Based Learning. International Journal of Continuing Engineering Education and Lifelong Learning 11(1/2), 25–34 (2001)
20. Brusilovsky, P.: Adaptive Hypermedia. User Modeling and User-Adapted Interaction 11, 87–110 (2001)
21. Nwana, H.: User Modeling and user adapted interaction in an intelligent tutoring system. User Modeling and User-Adapted Interaction 1(1), 1–32 (1991)
22. Graf, S.: Adaptivity in Learning Management Systems Focusing On Learning Styles. Unpublished Ph. D. Thesis (2007)
23. Cha, H.J., Kim, Y.S., Park, S.H., Yoon, T.B., Jung, Y.M., Lee, J.-H.: Learning Style Diagnosis Based on User Interface Behavior for the Customization of Learning Interfaces in an Intelligent Tutoring System. In: Ikeda, M., Ashley, K.D., Chan, W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 513–524. Springer, Heidelberg (2006)
24. Rovai, A.P., Barnum, K.T.: On-Line Course Effectiveness: An Analysis of Student Interactions and Perceptions of Learning. Journal of Distance Education 18(1), 57–73 (2003)

Exploring the State of the Art in Adaptive Distributed Learning Environments

Birol Ciloglugil[1] and Mustafa Murat Inceoglu[2]

[1] Ege University, Department of Computer Engineering,
35040 Bornova, Izmir, Turkey
[2] Ege University, Department of Computer Education and Instructional Technology,
35040 Bornova, Izmir, Turkey
{birol.ciloglugil,mustafa.inceoglu}@ege.edu.tr

Abstract. The use of one-size-fits-all approach is getting replaced by the adaptive, personalized perspective in recently developed learning environments. This study takes a look at the need of personalization in e-learning systems and the adaptivity and distribution features of adaptive distributed learning environments. By focusing on how personalization can be achieved in e-learning systems, the technologies used for establishing adaptive learning environments are explained and evaluated briefly. Some of these technologies are web services, multi-agent systems, semantic web and AI techniques such as case-based reasoning, neural networks and Bayesian networks used in intelligent tutoring systems. Finally, by discussing some of the adaptive distributed learning systems, an overall state of the art of the field is given with some future trends.

Keywords: Adaptive E-Learning Systems, Distributed Learning Environments, Intelligent Tutoring Systems, Multi-Agent Systems, Semantic Web.

1 Introduction

Nowadays most of the universities deliver online courses and/or offer distance learning programs. With distance learning programs, it is possible for a learner to access whichever course content he/she seeks, whenever he wants and wherever he/she is. Distance learning gives the opportunity of studying without time and place constrictions to the learners. Learning anytime and anywhere is a great advantage over the classic classroom-based education but there are disadvantages of distance learning as well. The learner may feel lonely because of the need of being socialized, he/she may have questions to ask to the instructors and may want to communicate and collaborate with other learners and teachers. Most of those needs are not satisfied in the early distance learning applications which consist of making the course materials online via a web page. With the recent technological developments, most of the e-learning systems allow their users to communicate both synchronously and asynchronously through virtual classrooms and message boards with other students and instructors.

With the new technological developments, new trends are emerging in the education domain and they are influencing e-learning systems with some shifts in the system design as followed [1]:

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 556–569, 2010.

- The shift from studying to graduate to studying to learn: Most learners are working and have well-defined personal goals for enhancing their careers.
- The shift from student to learner: This shift has resulted in a change in strategy and control, so that the learning process is becoming more cooperative than competitive.
- The shift from expertise in a domain to teaching beliefs: The classical teaching systems refer to "domain and teaching expertise" when dealing with the knowledge-transfer process, but the new trend is based on the concept of "belief." One teacher may have different beliefs from another and the different actors in the system (students, peers and teachers) may have different beliefs about the domains and teaching methods.
- The shift from a four-year program to graduate to lifelong learning: Most e-learners have a long-term learning plan related to their career needs.
- The shift to conceiving university departments as communities of scholars, but not necessarily in a single location.
- The shift to mobile learning: Most e-learners are working and have little spare time. Therefore, any computer-based learning must fit into their busy schedules (at work, at home, when traveling), so that they require a personal and portable system.

In order to support these emerging trends, e-learning systems are evolving with the key role of personalization. The distributed adaptive learning environments have two main features: adaptivity and distribution. With adaptivity, the system adapts to the personal characteristics and needs of each user. Adaptivity involves adaptive curriculum planning, adaptive sequencing, adaptive course generation, adaptive course delivery and adaptive testing. Adaptivity needs intelligence to mine databases of learner information and educational resources. Distribution, on the other hand, deals with the dynamic and distributed nature of data and applications in distributed learning environments.

Personalization in e-learning systems with its challenges and opportunities is discussed in the following section of the study. Section 3 explains some of the technologies that can be used to achieve personalization in adaptive distributed learning environments with a brief evaluation of these technologies. Finally, an overview of some of the available adaptive distributed learning systems is given in section 4.

2 Personalization in E-Learning

One of the main problems of the early distance learning systems is the one-size-fits-all approach. Every student has different characteristics such as his/her interests, goals and familiarity with the educational subject. Since their characteristics differ, a learning tool or a learning material can not be suitable for every learner. Learning materials (course contents and instruction methods) and learning tools (devices and interfaces) can get personalized with the help of the educational and instructional technologies for modeling the learner. The pedagogical aspect of personalization is beyond the aim of this study.

How medical students use various information and communication devices in the learning context is studied by [2] and it is stated that “there is no one-size-fits-all device that will suit all use situations and all users”. The multi-device paradigm fits well with the e-learning context, in which students use different devices, depending on the situation, environment and context. It is shown that personalized learning materials have increased the learning speed and helped learners achieve better understanding [3].

2.1 Why Personalization?

The early e-learning systems were designed with the client/server architecture where the students were modeled as the clients requesting appropriate course materials from the servers. This structure allows a limited level of personalization at the server side by a predefined classification of the students’ characteristics with the option of continuing evaluation through run time.

Most of the recent e-learning systems have departed the client/server architecture and are being designed as distributed systems. In distributed learning environments, the resources and the applications are dynamic and distributed in remote machines. It is expected from the application not only to respond to the requests for information but also to intelligently adapt to new conditions and seek ways to support the learners and instructors.

It is discussed in [4] that a distributed learning system must consider a decentralized approach in which overall management is performed centrally but course materials (hypertext and multimedia documents, technical manuals, scripts and other applications) are served up locally by using various pieces of software that run on students’ machines. Interactivity and intelligent tutoring capabilities (i.e., various help facilities) must be provided by client-side software, as well.

2.2 What Is Personalized?

An intelligent teaching system is commonly described in terms of a four-model architecture: the interaction model, the learner’s model, the domain expert and the pedagogical expert [5]. The interaction model is involved with the interface preferences and the presentation mode (text, image, sound, etc.) of contents. The learner model represents static beliefs about the learner and the learner’s learning style. The domain expert contains the knowledge about the domain concepts and the course components. The pedagogical expert contains the information on how to teach the course units to the individual learner. It contains two basic components: teaching strategies that define the teaching rules [6] and the diagnostic knowledge that defines the actions to take depending on the learner’s background, experience, interests and cognitive abilities [7]. Individualized courses can be generated and presented to the learner with the use of these components by course-content adaptation, course-navigation adaptation, learning strategy, interfaces and interaction.

Most of the four components discussed here put user modeling in the center of the adaptation process. Thus, an e-learning system’s behavior can be personalized only if the system has individual models of the learners. The interaction model also uses the device profile in addition to the user profile.

2.3 Challenges and Issues with the Development of Personalized Distributed Systems

Web-based distributed learning has the following issues to deal with. Software systems for distributed learning are typically complex, because they involve many dynamically interacting educational components, each with its own need for resources and involve in complex coordination. The systems have to be scaleable and accommodate networking, computing and software facilities that support thousands of simultaneous distributed users using different operating systems that can concurrently work and communicate with each other and receive adequate quality of service support [8].

Most of the existing web-based learning management systems are not concerned with individual learner differences and do not adjust to the profiles of individual students regarding actual skills, preferences, etc. Curriculums are designed just for a specific segment of the potential student population. Courses are build around textbooks and other materials designed for that curriculum and do not understand students' situations and requirements and do not utilize the possible contributions that students can make to the learning content and process. However, in personalized learning systems, course developers have the difficult task to generate personalized course materials. As the number of distributed learners increase, serious efficiency problems in course development and maintenance will occur [9].

3 How to Achieve Personalization in E-Learning Systems?

In order to achieve personalization in e-learning systems, a couple of technologies can be used as a standalone technology or as a hybrid approach combined with other technologies. Some of these technologies and their usage for personalization are explained below.

3.1 Web Services

A "web service" is an accessible application that other applications and humans can automatically discover and invoke. An application is a web service if it is (a) independent as much as possible from specific platforms and computing paradigms; (b) developed mainly for inter-organizational situations rather than for intra-organizational situations; and (c) easily able to be integrated (i.e., combining it with other web services does not require the development of complex adapters) [10]. Web services support a service-oriented view of distinct and independent software components interacting to provide valuable functionality.

Web services technologies have several limitations [11]: a web service knows only about itself, not about its users, clients, or customers; web services are not designed to use and reconcile ontologies among each other or with their clients; web services are passive until invoked and cannot provide alerts or updates when new information becomes available; and web services do not cooperate with each other or self-organize, although they can be composed by external systems.

3.2 Intelligent Agents

Agents are software programs that operate autonomously when triggered and perform tasks of repetitive nature. A Multi-Agent System (MAS) is described as “a loosely-coupled network of problem solvers that work together to solve problems that are beyond their individual capabilities” [12].

Agents have two main advantages in the distributed learning environments context: First, distributing tasks to numerous specialized, fine-grained agents promotes the modularity, flexibility and incrementality of learning systems and lets new services come and go without disturbing the overall system. The agents have their local knowledge about specific tasks and their autonomy. Limiting the complexity of an individual agent simplifies control, promotes reusability and provides a framework for tackling interoperability. Second, because of the agents’ autonomous nature, they have the ability to independently carry out tasks delegated to them by people or other software and this reduces the workload of users [9].

3.3 Semantic Web

The semantic web aims at adding semantic information to web contents in order to create an environment in which software agents will be capable of doing tasks efficiently [13].

From the perspective of course materials design, the course materials can be semantically annotated to be reused efficiently in other courses. Furthermore, this can facilitate the user access to their preferred contents. This semantic search and browsing is possible by the use of the ontological technology.

The semantic web can be considered a suitable platform for implementing e-learning systems, because they provide mechanisms for developing ontologies for learning, the semantic annotation of materials, their combination to define courses and its evaluation.

3.4 Artificial Intelligence (AI)

AI techniques are frequently used in the e-learning domain. There are three main intelligent competencies in MAS-based DLEs (Distributed Learning Environments): intelligent decision-making support, coordination and collaboration of the agents in the MAS and student modeling for personalization and adaptation in learning systems [9].

It is desirable for MAS-based distributed learning environments to provide more smart or intelligent learning functions that offer personalized services with capabilities to learn, reason, have autonomy and be totally dynamic. With intelligent learning environments, different educational institutions can collaborate to share education resources and manage them effectively. To this end, it is critical to embed intelligence in MAS-based DLEs, in other words, to develop human-like intelligent agents by applying artificial intelligence techniques such as case-based reasoning, symbolic machine learning and rule-based reasoning. An intelligent agent that performs teaching, learning or administration tasks on behalf of teachers, learners or administrators is a set of independent software tools or applications that communicates with other applications or agents within one or several computer environments [14].

3.5 Evaluation of the Usage of Personalization Technologies in E-Learning Systems

As a result of delivering a lot of advanced functionalities and being dynamic and distributed in nature, the complexity and the heterogeneousness of distributed learning systems bring serious issues to overcome. Agents can be used as the core components in intelligent distributed learning environments because of their inherent natures: autonomous, intelligent, sociable, etc. However, there are some challenges about the integration of agents into the existing learning environments or into heterogeneous learning environments. Since web services technology is characterized by its standardized communication protocol, interoperability, easy integration and easy deployment, it is an excellent complimentary partner with intelligent agents in distributed learning environments. Agent-supported web services in designing and developing distributed learning environments is discussed in [9] and it is expressed that the integration of web services and agents simplifies the complexity of development and makes distributed learning environments feasible and practical.

The agent technology has several advantages for implementing distributed systems. Agents enable the functionalities of the developed system to be distributed in small, reproducible and distributed software entities. It also allows for a clear and easy separation between their internal, private knowledge and their interface toward the external world and other agents. Semantic web provides ontologies for the efficient representation of agents' knowledge. AI techniques are used for the coordination and collaboration of the agents in the MAS to provide a better interface to the other agents and the external world. Thus, intelligent agents, web services, semantic web and AI can be used together to build more efficient and more intelligent MAS-based distributed learning environments.

4 Overview of Some Adaptive Distributed Learning Systems

In this section, some of the available adaptive distributed learning systems will be discussed. IEEE's learning technology systems architecture (LTSA) provides a model for studying learning environments. LTSA is generic enough to represent a variety of different learning systems from different domains. Fig. 1 shows the model used by IEEE's LTSA [15].

In the model, the learner entity is an abstraction of a human learner. The learner entity receives the final multimedia presentation, while the learner's behavior is observed and learning preferences are communicated with the coach. Then, the coach sends queries to the learning-resources component to search for learning content appropriate for the learner entity. The queries specify search criteria based, in part, on learning preferences, assessments and performance information. The appropriate locators (e.g., learning plans) are sent to the delivery process. The learning-resources component stores "knowledge" as a resource for the learning experience and the queries can be searched in this repository. The components to the right of the learner entity correspond to performance control. Performance is measured by the evaluation component and the measurements are stored in the records database. The coach, when locating new content, can then use the data in the records database.

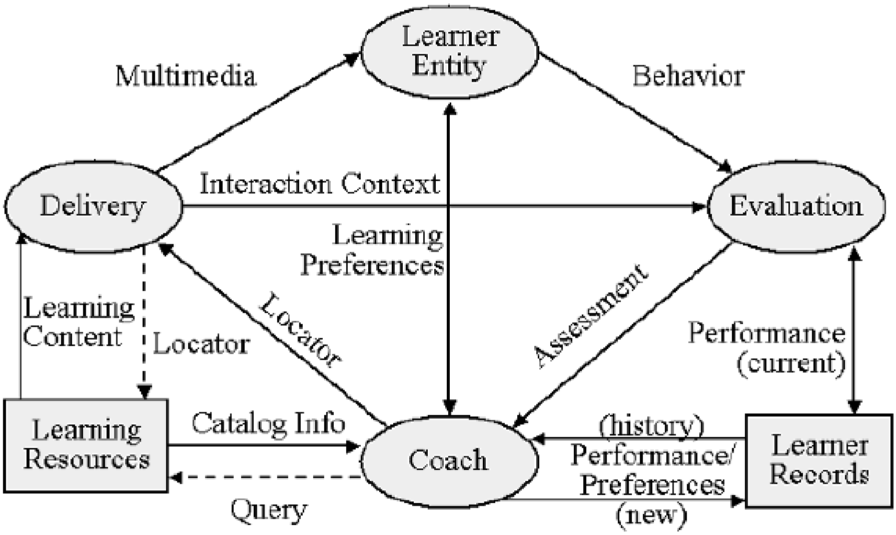


Fig. 1. IEEE's LTSA [15]

The design and implementation of a distributed learning resource registry system is illustrated in [16]. This system faces the challenge that e-learning users can hardly find the learning resources they look for, in the current distributed and relatively isolated environment where web-based learning contents are stored in different resource repositories or management systems. This study also depicts the "Distributed Learning Resource Registry and Discovery Model", which the implemented system is based on. The system enables developers and repository systems to register learning resources to the registry system and provides a discovery mechanism to find the requested learning resources. It is discussed that it can realize the integration of different learning resource repository or management systems in a loose-coupled manner by utilizing web service technology and building a distributed virtual learning resource marketplace.

In learning systems, a learning services architecture and learning services stack have been proposed by the Learning Systems Architecture Lab at Carnegie Mellon University. This provides a framework for developing service-based learning technology systems. In this approach, rather than building large, closed systems, the focus is on flexible architectures that provide interoperability of components and learning content and that rely on open standards for information exchange and component integration [17].

Intelligent tutoring systems usually consist of a domain module, a student model and a tutorial or pedagogical module [18, 19]. Fig. 2 shows the components of an intelligent tutoring system for distributed learning. As shown in the figure, for an intelligent tutoring system to be effective, the components must interact with each other by passing information between each other and learning from each other. An effective interface module is critical for allowing the student to interact from a distance with the different components of the intelligent tutoring systems and for getting the learner's attention and engaging the learner in the learning process.

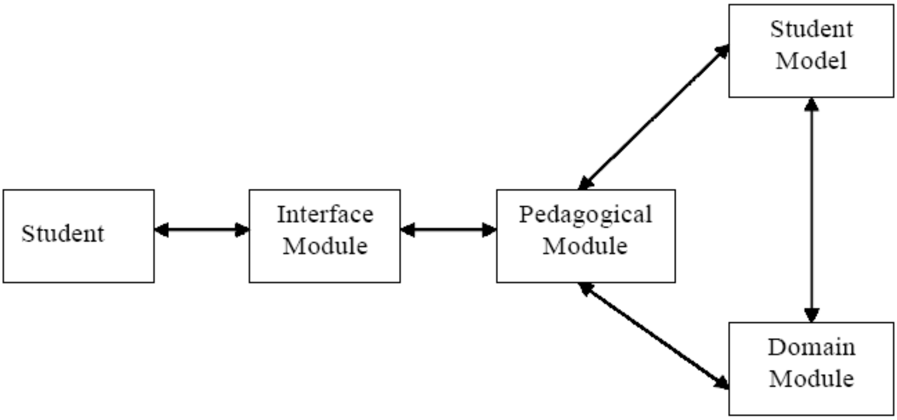


Fig. 2. Components of an intelligent tutoring system [9]

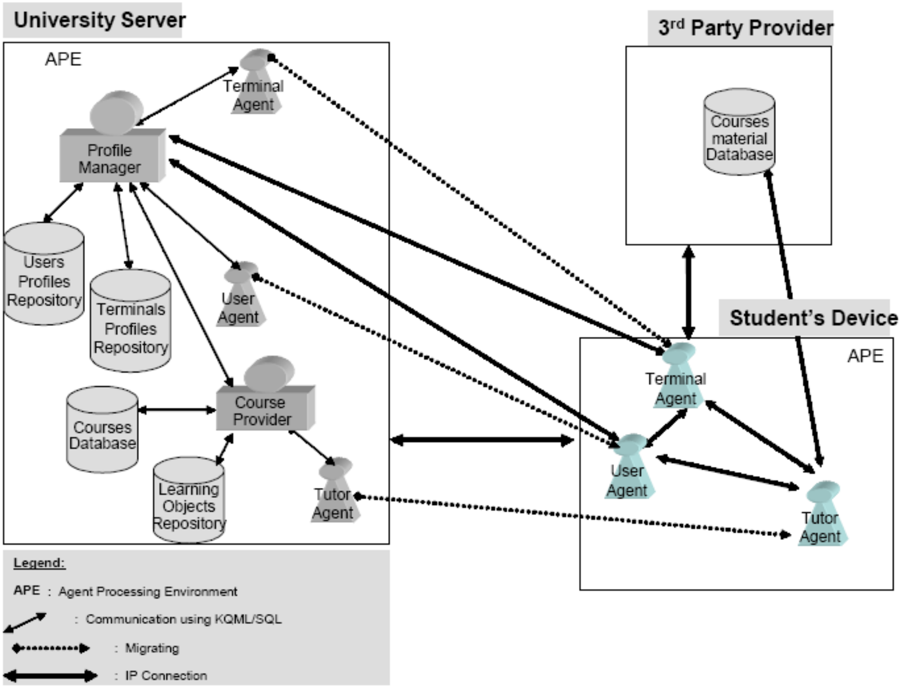


Fig. 3. The architecture of the proposed system in [9]

The agent technology could also be applied to create distributed Learning Management Systems (LMS). With some artificial intelligence methods it would be possible to build a distributed Intelligent Tutoring System (ITS). It is presented in [20] that there are two possible architectures of distributed LMS: One using an LMS to store the knowledge (such as a domain module of a ITS) and agents representing the tutor and student modules of an ITS. In the second approach, there is no LMS and all ITS functions are distributed between agents. The first approach contains Student Agent (SA), Professor Agent (PA), Manager Agent (MA) and Database Agent (DA); while the second approach contains Student Agent (SA), Professor Agent (PA) and School Manager (SM).

In [9], a multi-agent architecture is proposed for implementing an e-learning system that offers course personalization and supports mobile users connecting from different devices. The detailed architecture of the system contains five main components that are presented in Fig. 3:

1. User profile repository: For each user, the system maintains a profile that has two components: the learner's model and the user's preferences regarding learning style, interfaces and content display.
2. Device profile repository: For each device, the system maintains a profile of the features and capabilities useful for providing the e-learning service (screen size, bandwidth limit, colors, resolution, etc.). Some features that can be automatically detected by the system (operating system, browser, plug-ins) are not stored in the repository but are integrated with the profile when initializing the terminal agent.
3. Learning object repository: This contains the course's teaching material defined as learning objects [21].
4. Course database: For each course, the system maintains two knowledge structures: the course study guide and the course study plan.
5. The multi-agent system is composed of stationary and mobile agents.

In the MAS-based DLE proposed in [9], profile manager and course provider agents are the stationary agents running at the university server, while the user, terminal and tutor agents are the mobile agents initiated at the server and migrated to the corresponding user device. However, in the case of large agent communities, the tasks of the agents running both at the student and the university side can get significantly heavy, even more if the student agents run on devices with limited capabilities. To overcome this issue, a new multi-agent learning system called ISABEL is proposed in [22]. ISABEL contains the device and the student agents representing the student and its device respectively. There are also the teacher and the tutor agents running at the university side of the system. The basic idea underlying ISABEL is partitioning the students in clusters of students that have similar profiles, where each cluster is managed by a tutor agent. Consequently, when a student accesses the e-learning site, the teacher agent gives control to the tutor agent associated with the cluster which the student belongs to. With this design, the work loads of the teacher agents are lowered significantly.

The use of ontologies to model the knowledge of specific domains represents a key aspect for the integration of information coming from different sources, for supporting collaboration within virtual communities, for improving information retrieval and more generally, it is important for reasoning on available knowledge. In the e-learning field, ontologies can be used to model educational domains and to build, organize and update specific learning resources (i.e. learning objects, learner profiles, learning paths, etc.). One of the main problems of educational domains modeling is the lacking of expertise in the knowledge engineering field by the e-learning actors. An advanced ontology management system for personalized e-learning is given in [23]. It presents an integrated approach to manage the life-cycle of ontologies, used to define personalized e-learning experiences supporting blended learning activities, without any specific expertise in knowledge engineering.

An application of intelligent techniques and semantic web technologies in e-learning environments is discussed and a semantic web technologies-based multi-agent system that allows to automatically controlling students' acquired knowledge in e-learning frameworks is developed in [24]. According to this study, the essential elements of effective learning are control of students' skills and feedback between students and their tutor. The main idea behind the approach presented is that a domain ontology is not only useful as a learning instrument but it can also be employed to assess students' skills. For it, each student is prompted to express his/her beliefs by building her/his own discipline-related ontology and then it is compared to a reference one. The analysis of students' mistakes allows to propose them personalized recommendations and to improve the course materials in general.

In [25] a machine learning based learner modeling system for adaptive web-based learning is proposed. It is discussed that a web-based learning system should be in multi-layered sense as depicted in Fig. 4 which contains the learning management system (LMS) layer, the learner modeling system layer and the user interface. The learner modeling system layer acts as a mediator between the user interface and learning management system layers. Some of the frequently used AI techniques for learner modeling are neural networks, fuzzy systems, nearest neighbor algorithms, genetic algorithms, Bayesian networks and hybrid systems which consist of the combinations of different techniques. The proposed learner modeling system has a three-layered architecture in which Bayesian networks, fuzzy systems and artificial neural networks are used together to compose a hybrid system.

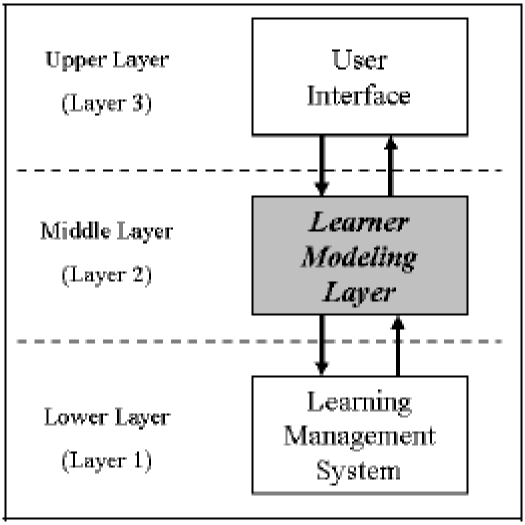


Fig. 4. Layered structure of the web-based learning system proposed in [25]

A reactive architecture for ambient e-learning is given in [26]. As discussed in [27, 28] the main challenges of ambient intelligence in e-learning systems are: transparent integration into the environment (adaptation of the presented material, repetition of certain aspects of a lesson and integration of exercises should be possible in different contexts), adaptive software platform (the learner should be able to take the course with him wherever he happens to move, this requires the underlying software to be highly adaptive), perception of the environment (observation of the user to get information about his current state via eye-tracker, CCD cameras and sensors to be used for mimic and gesture recognition), multimodal interaction (depending on the learning style of the user, the system needs to use different modalities alone or in combination to achieve the most success) and learning and adaptation (the system should be able to adapt to a user's learning style; if it didn't manage to teach a certain concept with the approach it planned on at first, it needs to be able to find alternative routes to achieving it's teaching goal).

Some of the key tasks for ambient e-learning are [26]: Sensor data processing (continuous aggregation of the sensory data into a dynamic environment and user model), Didactic guidance (determination of learning objectives and context and goal specific selection of primitive learning units) and Dynamic course composition (generation of structured and adaptive learning workflows from elementary learning units). An e-learning architecture proposed in [26], where each of these tasks is handled by a specific module is presented in Fig. 5.

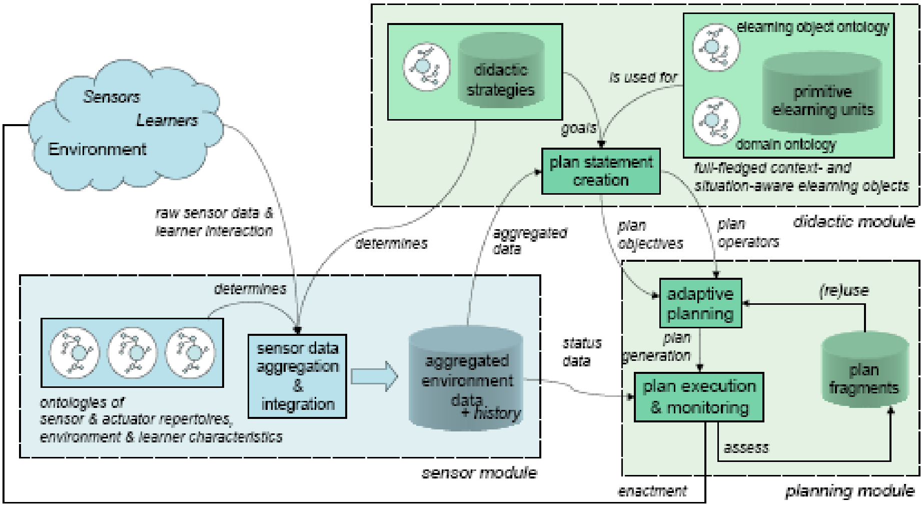

Sensors
Learners
Environment
didactic strategies
is used for
elearning object ontology
primitive elearning units
domain ontology
goals
plan statement creation
full-fledged context- and situation-aware elearning objects
raw sensor data & learner interaction
determines
aggregated data
plan objectives
plan operators
didactic module
determines
sensor data aggregation & integration
aggregated environment data + history
ontologies of sensor & actuator repertoires, environment & learner characteristics
status data
adaptive planning
(re)use
plan generation
plan execution & monitoring
plan fragments
assess
sensor module
enactment
planning module

Fig. 5. Architecture of an ambient e-learning system given in [26]

5 Conclusion

In this paper, the personalization issues in e-learning systems and the technologies that are used for designing and developing adaptive distributed learning environments have been discussed. Some of the challenges encountered are; the complexity and the heterogeneousness of the distributed learning environments, the workloads of users (educators and learners) with personalized assistance when resources are widely distributed, heterogonous and ever-changing, the integration of agents into the existing learning environments or into heterogeneous learning environments and the lack of methodology for systems modeling, in particular, knowledge modeling.

Web services technology focuses on flexible architectures that provide interoperability of components and learning contents and relies on open standards for information exchange and component integration. Web services are used for integrating existing systems and for accessing data in heterogeneous environments. Agents can act as both the requester and the provider of web services. As a requester, an agent can perform searches of different web services and calls to web services. As a provider, an agent has a dual nature that combines the characteristics of the two technologies: the ability to be published, found and called as a web service and the ability to make autonomous decisions. The integration of agents and web services provide an efficient mechanism for developing distributed learning environments.

There are two aspects to the obstacles of using agent technology in distributed learning systems; the difficulty of understanding and interacting with data and the agent knowledge modeling. Knowledge modeling can be characterized as a set of techniques that focus on the specification of static and dynamic knowledge resources. The semantics integration and knowledge management can also be addressed as active research areas, with the key role of ontology-based domain modeling.

To reduce the information workloads of educators and provide assistance to learners, distributed learning environments require that software not merely respond to requests for information but intelligently adapt and actively seek ways to support learners and educators. Applying AI technologies can make e-learning systems personalized, adaptive and intelligent. Although AI techniques have been used successfully in some e-learning systems, they have not yet been adopted in widely used e-learning systems, especially the open-source LMSs like Moodle. As a result, current intelligent LMSs are still in their early stage. Using AI techniques to generate more intelligent agents and to achieve more intelligent systems are ongoing research topics in distributed learning environments.

Implementation of user-side device independence for web contents is another subject that the researchers are working on. The aim in the future is to implement an infrastructure (a virtual learning resource e-market place) to provide collaborative e-learning services.

References

1. Kay, J.: Learner control. User Modeling and User-Adapted Interaction 11, 111–127 (2001)
2. Gallis, H., Kasbo, J.P., Herstad, J.: The multidevice paradigm in know-mobile - Does one size fit all? In: Bjørnestad, S., Moe, R.E., Mørch, A.I., Opdahl, A.L. (eds.) Proceedings of the 24th Information System Research Seminar in Scandinavia, pp. 491–504 (2001)
3. Brusilovsky, P., Vassileva, J.: Course sequencing techniques for large-scale web-based education. International Journal of Continuing Engineering Education and Lifelong Learning 13(1,2), 75–94 (2003)

4. Wang, H., Holt, P.: The design of an integrated course delivery system for Web-based distance education. In: Proceedings of the IASTED International Conference on Computers and Advanced Technology in Education (CATE 2002), pp. 122–126 (2002)
5. Wenger, E.: Artificial intelligence and tutoring systems. In: Computational and cognitive approaches to the communication of knowledge, pp. 13–25. Morgan Kaufmann, Los Altos (1987)
6. Vassileva, J.: A new approach to authoring of Adaptive Courseware for Engineering domains. In: Proceedings of the International Conference on Computer Assisted Learning in Science and Engineering (CALISCE 1994), pp. 241–248 (1994)
7. Specht, M., Oppermann, R.: ACE-Adaptive Courseware Environment. The New Review of Hypermedia and Multimedia 4, 141–161 (1998)
8. Vouk, M.A., Bitzer, D.L., Klevans, R.L.: Workflow and end-user quality of service issues in Web-based education. IEEE Trans. on Knowledge and Data Engineering 11(4), 673–687 (1999)
9. Lin, F.O.: Designing Distributed Learning Environments with Intelligent Software Agents. Information Science Publishing (2004)
10. Dale, J., Ceccaroni, L., Zou, Y., Agam, A.: Implementing agent-based Web services, challenges in open agent systems. In: Autonomous Agents and Multi-Agent Systems Conference in Melbourne, Australia, July 14–17 (2003)
11. Huhns, M.N.: Agents as Web Services. IEEE Internet Computing, 93–95 (July/August 2002)
12. O'Hare, G.M.P., Jennings, N.R.: Foundations of distributed artificial intelligence. John Wiley & Sons, New York (1996)
13. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American, 34–43 (2001)
14. Jafari, A.: Conceptualizing intelligent agents for teaching and learning. Educause Quarterly (3), 28–34 (2002)
15. IEEE LTSC, http://ltsc.ieee.org
16. Gang, Z., ZongKai, Y., Kun, Y.: Design and implementation of a distributed learning resource registry system. In: The Fourth International Conference on Computer and Information Technology CIT 2004, pp. 333–338 (2004)
17. Blackmon, W.H., Rehak, D.R.: Customized learning: A Web services approach. In: Proceedings: Ed-Media (2003)
18. Ong, J., Ramachandran, S.: Intelligent tutoring systems: The what and the how, http://www.learningcircuits.org/feb2000/ong.html
19. Thomas, E.: Intelligent tutoring systems, http://coe.sdsu.edu/eet/Articles/tutoringsystem/start.htm
20. Maia, R.F., Netto, M.L.: Work in Progress - A Distributed Approach to a Learning Management System using Multi-Agent Technology. In: Frontiers in Education (2005)
21. Wiley, D.A.: Connecting learning objects to instructional design theory. In: Wiley, D.A. (ed.) A definition, a metaphore and a taxonomy (2000)
22. Garruzzo, S., Rosaci, D., Sarne, G.M.L.: ISABEL: A Multi Agent e-Learning System That Supports Multiple Devices. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 85–88 (2007)
23. Gladun, A., Rogushina, J., Garcıa-Sanchez, F., Martínez-Béjar, R., Fernández-Breis, J.T.: An application of intelligent techniques and semantic web technologies in e-learning environments. Expert Syst. Appl. 36(2), 1922–1931 (2009)
24. Gaeta, M., Orciuoli, F., Ritrovato, P.: Advanced ontology management system for personalised e-Learning. Know-Based Syst. 22(4), 292–301 (2009)

25. Aslan, B.G., Inceoglu, M.M.: Machine Learning Based Learner Modeling for Adaptive Web-Based Learning. In: Gervasi, O., Gavrilova, M.L. (eds.) ICCSA 2007, Part I. LNCS, vol. 4705, pp. 1133–1145. Springer, Heidelberg (2007)
26. Beckstein, C., Denzler, J., Fothe, M., König-Ries, B., Sack, H., Vogel, J.: A Reactive Architecture for Ambient E-Learning. In: Proc. of Towards Ambient Intelligence: Methods for Cooperating Ensembles in Ubiquitous Environments (2007)
27. Shadbolt, N.: Ambient intelligence. IEEE Intelligent Systems 18(4), 2–3 (2003)
28. Paraskakis, I.: Ambient learning: a new paradigm for e-learning. In: Proc. 3rd Int. Conf. on multimedia and Information & Communication Technologies in Education (m-ICTE 2005), Recent Research developments in Learning Technologies, Caceres, Spain (2005)

Analysis of an Implicitly Restarted Simpler GMRES Variant of Augmented GMRES

Ravindra Boojhawon, Desire Yannick Tangman, Kumar Dookhitram,
and Muddun Bhuruth*

Department of Mathematics,
University of Mauritius, Reduit, Mauritius
{r.boojhawon,mbhuruth,y.tangman,kdookhitram}@uom.ac.mu
http://www.uom.ac.mu

Abstract. We analyze a Simpler GMRES variant of augmented GMRES with implicit restarting for solving nonsymmetric linear systems with small eigenvalues. The use of a shifted Arnoldi process in the Simpler GMRES variant for computing Arnoldi basis vectors has the advantage of not requiring an upper Hessenberg factorization and this often leads to cheaper implementations. However the use of a non-orthogonal basis has been identified as a potential weakness of the Simpler GMRES algorithm. Augmented variants of GMRES also employ non-orthogonal basis vectors since approximate eigenvectors are added to the Arnoldi basis vectors at the end of a cycle and in case the approximate eigenvectors are ill-conditioned, this may have an adverse effect on the accuracy of the computed solution. This problem is the focus of our paper where we analyze the shifted Arnoldi implementation of augmented GMRES with implicit restarting and compare its performance and accuracy with that based on the Arnoldi process. We show that augmented Simpler GMRES with implicit restarting involves a transformation matrix which leads to an efficient implementation and we theoretically show that our implementation generates the same subspace as the corresponding GMRES variant. We describe various numerical tests that indicate that in cases where both variants are successful, our method based on Simpler GMRES keeps comparable accuracy as the augmented GMRES variant. Also, the Simpler GMRES variants perform better in terms of computational time required.

Keywords: Nonsymmetric linear systems; Augmented GMRES; Simpler GMRES; Implicit Restarted Arnoldi.

1 Introduction

It is well-known that the GMRES [15] algorithm with restarting encounters difficulties in solving nonsymmetric linear systems with small eigenvalues. This slow convergence can be overcome through the process of deflation of the Krylov

* Author of Correspondence.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 570–585, 2010.

subspace. Morgan [9,10,11] formulated three equivalent methods which all have the effect of annihilating the components of the residual vector corresponding to the eigenvalues of smallest magnitudes.

Simpler GMRES [17] is a variant of the GMRES algorithm which computes an orthonormal basis of a shifted Krylov subspace, leading to an upper triangular least squares problem which results in a cheaper implementation. However the method has been criticized because of the use of a non-orthogonal basis for extracting the approximate solution.

Two augmented Simpler GMRES variants have been derived. The SGMRES-E variant which adds harmonic Ritz vectors to obtain an augmented basis has been described in [1] and SGMRES-DR, the Simpler GMRES variant with deflated restarting is described in [2]. In this paper, we analyse the SGMRES algorithm with implicit restarting [16]. Augmented variants of GMRES employ a non-orthogonal basis since approximate eigenvectors are added to the Arnoldi vectors for computing the approximate solution and in case these eigenvectors are ill-conditioned, the accuracy of the extracted approximate solution may be affected. Our main finding is that this variant of Simpler GMRES is a comparable method to its corresponding GMRES variant if we require a reasonable degree of accuracy of up to 10^{-10}. For a higher accuracy, we remark that the Simpler GMRES variants exhibit similar phenomenon of the erroneous decrease of the updated residual norms as pointed out in [17].

An outline of this paper is as follows. In §2, we recall the fundamental Simpler GMRES relationship which is the basis for a cheaper implementation and we discuss the transformation matrix associated with the algorithm. In §3, we discuss an efficient implementation of the implicitly restarted Simpler GMRES which its corresponding transforming matrix and in §4, we show that our algorithm which uses the Shifted Arnoldi process also yields the same subspace as implicitly restarted GMRES [10]. Finally, in §5, we give some numerical results to show the convergence history of our new algorithm.

2 The Simpler GMRES Algorithm

We consider the iterative solution of the large sparse nonsymmetric linear system

$$Ax = b,\ A \in \mathbb{R}^{n\times n},\ x,\, b \in \mathbb{R}^n, \tag{1}$$

with an initial guess x_0. Starting with the initial vector $w_1 = Ar_0/\|Ar_0\|_2$ where $r_0 = b - Ax_0$ is the associated residual vector, restarted Simpler GMRES, denoted SGMRES(m), first constructs an orthonormal basis of $W_m = [w_1,\, w_2,\, \ldots,\, w_m]$ of the image $A\mathcal{K}_m(A;\, r_0)$ of the Krylov subspace $\mathcal{K}_m(A;\, r_0)$, which we assume to be of dimension m and is given by

$$\mathcal{K}_m(A;\, r_0) = \operatorname{span}\left\{r_0,\, Ar_0,\, \ldots,\, A^{m-1}r_0\right\}.$$

This construction leads to the fundamental Simpler GMRES decomposition

$$AY_m = W_m R_m, \tag{2}$$

where R_m is a nonsingular upper triangular matrix and Y_m given by

$$Y_m = [r_0, W_{m-1}], \tag{3}$$

is the Simpler GMRES basis used for extracting approximate solutions of (1). Writing the mth approximate solution x_m in the form $x_m = x_0 + Y_m\widehat{y}$ with $\widehat{y} \in \mathbb{R}^m$, we find that the residual vector r_m is given by

$$r_m = r_0 - W_m R_m \widehat{y}.$$

The orthogonal residual principle $r_m \perp A\mathcal{K}_m(A;\, r_0)$ means that $W_m^{\mathrm{T}} r_m = 0$ and thus $\widehat{y}$ solves the upper triangular system

$$R_m\widehat{y} = \widetilde{w}, \tag{4}$$

where

$$\widetilde{w} = W_m^{\mathrm{T}} r_0 = [\xi_1, \xi_2, \ldots, \xi_m]^{\mathrm{T}}.$$

Now consider the matrix $W_m^{\mathrm{T}} Y_m = T_m^{-1}$ and partition $\widetilde{w}$ in the form $\widetilde{w} = [s_{m-1}, \xi_m]^{\mathrm{T}}$ where $s_{m-1} = (\xi_1, \ldots, \xi_{m-1})^{\mathrm{T}}$. It then follows that

$$T_m^{-1} = \begin{pmatrix} s_{m-1} & I_{m-1} \\ \xi_m & 0 \end{pmatrix},$$

where I_m denotes an identity matrix of order m. Assume that $\xi_m \neq 0$ ($\xi_m = 0$ is equivalent to stagnation of the algorithm [4]) and note that $\det T_m^{-1} = (-1)^{m+1}\xi_m$. We thus find that the matrix T_m is given by

$$T_m = \begin{pmatrix} 0 & 1/\xi_m \\ I_{m-1} & -\Xi_{m-1}/\xi_m \end{pmatrix}, \tag{5}$$

where $\Xi_{m-1} = [\xi_1\, \xi_2\, \ldots\, \xi_{m-1}]^{\mathrm{T}}$.
It can be checked that the matrix T_m shifts the $(j-1)$th column e_{j-1} of the identity matrix I_m to e_j for $j = 2, 3, \ldots, m$ and it transforms $\widetilde{w}$ onto e_1. Another important property that we use later is that the matrices W_m and $Y_m T_m$ differ only in their last column. More precisely, it is easy to show that

$$Y_m T_m - W_m = [0, \ldots, 0, Y_m t_m - w_m],$$

where t_m given by (9) is the last column of the transformation matrix T_m.

An augmented variant of SGMRES requires the computations of harmonic Ritz vectors at the end of a cycle. Seeking a harmonic Ritz vector $\widetilde{u}$ in the form $\widetilde{u} = Y_m\widetilde{g} \in A\mathcal{K}_m(A;\, r_0)$ and using the orthogonal residual property gives

$$W_m^{\mathrm{T}}\left(AY_m\widetilde{g} - \tilde{\theta} Y_m\widetilde{g}\right) = 0.$$

Using (2), the above equation can be written in the form

$$T_m R_m \widetilde{g} = \tilde{\theta}\widetilde{g}. \tag{6}$$

Noting that $T_m(R_mT_m)T_m^{-1} = T_mR_m$, we find that, at the end of the first cycle, harmonic Ritz vectors can be computed by first solving the eigenvalue problem

$$\widetilde{R}_m\widehat{g} = \tilde{\theta}\widehat{g}, \tag{7}$$

where $\widetilde{R}_m = R_mT_m$ and $\widehat{g} = T_m^{-1}\widetilde{g}$. A harmonic Ritz vector $\widetilde{u}$ corresponding to an eigenvector $\widehat{g}$ is given by $\widetilde{u} = Y_m\widetilde{g} = Y_mT_m\widehat{g}$.

It remains to be seen that $\widetilde{R}_m = R_mT_m$ can be easily computed. First observe that, if

$$R_m = \begin{pmatrix} \varrho_{11} & \varrho_{12} & \cdots & \varrho_{1m} \\ & \varrho_{22} & \cdots & \varrho_{2m} \\ & & \ddots & \vdots \\ & & & \varrho_{mm} \end{pmatrix}, \tag{8}$$

then

$$\widetilde{R}_m = R_mT_m = \begin{pmatrix} \varrho_{12} & \varrho_{13} & \cdots & \varrho_{1m} & \varsigma_1 \\ \varrho_{22} & \varrho_{23} & \cdots & \varrho_{2m} & \varsigma_2 \\ & \varrho_{33} & \cdots & \varrho_{3m} & \varsigma_3 \\ & & \ddots & \vdots & \vdots \\ & & & \varrho_{mm} & \varsigma_m \end{pmatrix}.$$

We thus find that the computation of $\widetilde{R}_m$ only requires the computation of the m scalars ς_i, for $i = 1, 2, \ldots, m$ by evaluating the matrix-vector product R_mt_m where

$$\xi_mt_m = [1, -\Xi_{m-1}]^T. \tag{9}$$

Since A is nonsingular, R_m is nonsingular and since $\det R_m = \prod_{i=1}^m \varrho_{ii}$, we find that $\widetilde{R}_m$ is an unreduced upper Hessenberg matrix.

3 Implicitly Restarted Simpler GMRES

We study the implementation of the implicitly restarted variant of Simpler GMRES. Our starting point is the fundamental Simpler GMRES relation (2) which results after an initial run of SGMRES(m). Let $T_m = \left(W_m^{\mathrm{T}}Y_m\right)^{-1}$ denote the SGMRES transformation matrix given by (5). Postmultiplication of $AY_m = W_mR_m$ by T_m and denoting $\widetilde{R}_m = R_mT_m$ we obtain the relation

$$AY_mT_m = W_m\widetilde{R}_m. \tag{10}$$

The second step in SGMRES-IR consists of successively applying the implicit QR algorithm with the harmonic Ritz values $\tilde{\theta}_i$ for $i = k+1, \ldots, m$ as shifts. In the following, we assume that the $m-k$ largest harmonic Ritz values are real and arranged in increasing order.

For the shift $\tilde{\theta}_{k+1}$, consider a QR factorization of $\widetilde{R}_m$ in the form $(\widetilde{R}_m - \tilde{\theta}_{k+1}I_m) = Q^{(1)}R^{(1)}$. We then have

$$\begin{aligned} A(Y_mT_m) - \tilde{\theta}_{k+1}W_m &= W_m(\widetilde{R}_m - \tilde{\theta}_{k+1}I_m), \\ A(Y_mT_m) - \tilde{\theta}_{k+1}W_m &= W_mQ^{(1)}R^{(1)}, \\ A(Y_mT_m)Q^{(1)} - \tilde{\theta}_{k+1}W_mQ^{(1)} &= (W_mQ^{(1)})R^{(1)}Q^{(1)}, \\ A(Y_mT_m)Q^{(1)} &= (W_mQ^{(1)})(R^{(1)}Q^{(1)} + \tilde{\theta}_{k+1}I_m). \end{aligned} \tag{11}$$

Let $Y_m^{(1)} = Y_mT_mQ^{(1)}$, $W_m^{(1)} = W_mQ^{(1)}$ and $R_m^{(1)} = \left(R^{(1)}Q^{(1)} + \tilde{\theta}_{k+1}I_m\right) = \left(Q^{(1)}\right)^{\mathrm{T}} \widetilde{R}_mQ^{(1)}$. We can thus write the following decomposition

$$AY_m^{(1)} = W_m^{(1)}R_m^{(1)}.$$

Applying the matrices in (11) to the vector e_1, we find that the first vector $w_1^{(1)}$ in $W_m^{(1)}$ satisfies

$$w_1^{(1)} = \frac{1}{e_1^{\mathrm{T}}R_m^{(1)}e_1}\left(A - \tilde{\theta}_{k+1}I\right)w_1, \tag{12}$$

where $e_1^{\mathrm{T}}R_m^{(1)}e_1 \neq 0$ which is always ensured during the QR factorisation. Since Y_mT_m and W_m differ only in their last column and $Q^{(1)}$ is upper Hessenberg, the first $m-2$ columns of $Y_m^{(1)}$ and $W_m^{(1)}$ are identical.

After $m-k$ implicit shifts, we let $Q = Q^{(1)}Q^{(2)}\cdots Q^{(m-k)}$ where $Q^{(j)}$ denotes the orthogonal matrix associated with the shift $\tilde{\theta}_{k+j}$ for $j = 1, 2, \ldots, m-k$. We then obtain the decomposition

$$AY_m^{(m-k)} = W_m^{(m-k)}R_m^{(m-k)}, \tag{13}$$

where $Y_m^{(m-k)} = Y_mT_mQ$ and $W_m^{(m-k)} = W_mQ$. Equating the first k columns of (13) gives $AY_k^{(m-k)} = W_k^{(m-k)}R_k^{(m-k)}$. Denoting $Y_k^+ = Y_k^{(m-k)}$, $W_k^+ = W_k^{(m-k)}$ and $R_k^+ = R_k^{(m-k)}$, we obtain a length k Simpler Arnoldi type relation

$$AY_k^+ = W_k^+R_k^+. \tag{14}$$

Then, using similar arguments as for a single shift, we can deduce the following result.

Lemma 1. *The first $k-1$ columns of Y_k^+ and W_k^+ in (14) are identical and*

$$w_1^+ = \bar{\alpha}\prod_{l=1}^{m-k}\left(A - \tilde{\theta}_{k+l}I\right)w_1,$$

where $\bar{\alpha}$ is a normalizing factor.

We then extend (14) to length m using Algorithm 1 given below.

Algorithm 1. Simpler Arnoldi Extension

```
Input: $A$, $r_0$, $W_k := W_k^+$, $Y_k := Y_k^+$, $R_k := R_k^+$;
Output: $W_m$, $Y_m$, $R_m$;
(1). for $i = k+1,\ k+2,\ \dots,\ m$ do
        if $i = k+1$, $w_i := Ar_0$;
        else $w_i := Aw_{i-1}$; endif
        for $j = 1,\ 2,\ \dots,\ i-1$ do
            $\varrho_{ji} := (w_j,\ w_i)$; $w_i := w_i - \varrho_{ji} w_j$;
        endfor $j$
        $\varrho_{ii} := \|w_i\|_2$; $w_i := w_i / \varrho_{ii}$;
    endfor $i$
(2). Output: $Y_m := [Y_k,\ r_0,\ w_{k+1},\ \dots, w_{m-1}]$; $W_m := [W_k,\ w_{k+1},\ \dots,\ w_m]$; $R_m$;
```

Fig. 1 shows the change in the matrix structure of R_m following the implicit QR and the extension phases.

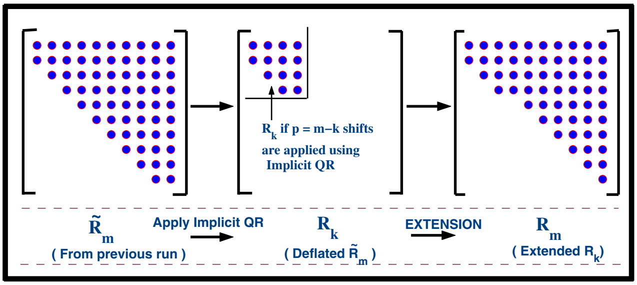


Fig. 1. Implicit QR and extension to m

The SGMRES-IR approximate solution is given by $x_m = x_0 + z_m$ where now

$$z_m = \left[Y_k^+,\ r_0,\ w_{k+1},\ \dots, w_{m-1}\right] \widehat{y}.$$

Requiring that $W_m^{\mathrm{T}} r_m = 0$ is again equivalent to $R_m \widehat{y} = W_m^{\mathrm{T}} r_0$. Now partitioning $W_m^T r_0$ in the form

$$W_m^T r_0 = \left[\widetilde{w}^{(k)},\ \widetilde{w}^{(m-k)}\right]^T,$$

where the vectors $\widetilde{w}^{(k)}$ and $\widetilde{w}^{(m-k)}$ are of length k and $m-k$ respectively, we find that $\widetilde{w}^{(k)} = 0$. Thus the correction z_m can be computed by first solving the linear system

$$\begin{pmatrix} R_k & \widehat{R} \\ 0 & R_{m-k} \end{pmatrix} \begin{pmatrix} \widehat{y}^{(k)} \\ \widehat{y}^{(m-k)} \end{pmatrix} = \begin{pmatrix} 0 \\ \widetilde{w}^{(m-k)} \end{pmatrix}, \tag{15}$$

where the matrices $\widehat{R}$ and R_{m-k} have dimensions $k \times (m-k)$ and $(m-k) \times (m-k)$ respectively. Since R_{m-k} is upper triangular, we can easily solve (15) by first solving the upper triangular system

$$R_{m-k} \widehat{y}^{(m-k)} = \widetilde{w}^{(m-k)},$$

and then solving the $(k \times k)$ upper-Hessenberg system

$$R_k \widehat{y}^{(k)} = -\widehat{R}\widehat{y}^{(m-k)}.$$

Since the first $k-1$ columns of Y_m and W_m are identical, we find that if we denote $\widehat{y} = [\widehat{\eta}_1, \widehat{\eta}_2, \ldots, \widehat{\eta}_m]^{\mathrm{T}}$, then the SGMRES-IR correction z_i can be obtained from

$$z_i = \begin{cases} (w_1, \ldots, w_i)\,\widehat{\eta}_i, & \text{if } 1 \le i \le k-1, \\ (w_1, \ldots, w_{k-1}, y_k)\,\widehat{\eta}_i, & \text{if } i = k, \\ (w_1, \ldots, w_{k-1}, y_k, r_0)\,\widehat{\eta}_i, & \text{if } i = k+1, \\ (w_1, \ldots, w_{k-1}, y_k, r_0, w_{k+1}, \ldots, w_{i-1})\,\widehat{\eta}_i, & \text{if } k+1 < i \le m\,. \end{cases} \tag{16}$$

The residual vector r_m and the residual norm update $\|r_i\|_2$ for $i = k+1, \ldots, m$ can be obtained using

$$r_m = r_0 - W_m R_m \widehat{y}. \tag{17}$$

and

$$\|r_i\|_2 = \sqrt{\|r_{i-1}\|_2^2 - \xi_i^2}. \tag{18}$$

respectively.

3.1 SGMRES-IR Transformation Matrix

Let $W_m^{\mathrm{T}} Y_m = T_m^{-1}$ and partition $W_m = [W_k, W_{m-k}]$ and $Y_m = [Y_k, Y_{m-k}]$ into two blocks of column vectors of length, k and $m-k$ respectively. Then

$$W_m^T Y_m = \begin{pmatrix} W_k^T Y_k & W_k^T Y_{m-k} \\ W_{m-k}^T Y_k & W_{m-k}^T Y_{m-k} \end{pmatrix}.$$

Lemma 1 implies that $W_k^T Y_k = I_k$ and from the othogonalisation process carried out in Algorithm 3.2, we have $W_k^T Y_{m-k} = 0$. Next, consider the term $W_{m-k}^{\mathrm{T}} Y_k$ given by

$$W_{m-k}^{\mathrm{T}} Y_k = \begin{pmatrix} w_{k+1}^{\mathrm{T}} \\ w_{k+2}^{\mathrm{T}} \\ \vdots \\ w_m^{\mathrm{T}} \end{pmatrix} [w_1, w_2, \ldots, w_{k-1}, y_k]\,.$$

Letting $\tau_k = w_m^{\mathrm{T}} y_k$, we find that

$$w_j^{\mathrm{T}} y_k = \frac{\xi_j}{\xi_m} \tau_k, \quad k+1 \le j \le m-1,$$

and thus

$$\widehat{T} = W_{m-k}^{\mathrm{T}} Y_k = \left[0, \widetilde{t}_{m-k}\right],$$

where

$$\widetilde{t}_{m-k} = \left[\frac{\xi_{k+1}}{\xi_m}\tau_k,\, \ldots,\, \frac{\xi_{m-1}}{\xi_m}\tau_k,\, \tau_k\right]^{\mathrm{T}}.$$

Finally note that

$$T_{m-k}^{-1} = W_{m-k}^{T} Y_{m-k} = \begin{pmatrix} \hat{s}_{m-1} & I \\ \xi_m & 0 \end{pmatrix}$$

where $\hat{s}_{m-1} = (\xi_{k+1},\, \xi_{k+2},\, \ldots, \xi_{m-1})^{\mathrm{T}}$. We therefore find that

$$T_m^{-1} = \begin{pmatrix} I_k & 0 \\ \widehat{T} & T_{m-k}^{-1} \end{pmatrix}.$$

Now writing the SGMRES-IR transformation matrix T_m in the form

$$T_m = \begin{pmatrix} I_k & 0 \\ \widetilde{T} & T_{m-k} \end{pmatrix},$$

it is easy to see that

$$T_{m-k} = \begin{pmatrix} 0 & 1/\xi_m \\ I_{m-1} & -\frac{1}{\xi_m}\hat{s}_{m-1} \end{pmatrix}$$

and that the matrix $\widetilde{T}$ has a single non-zero entry $-\tau_k/\xi_m$ and is given by

$$\widetilde{T} = -T_{m-k}\widehat{T} = \begin{pmatrix} 0 & -\frac{\tau_k}{\xi_m} \\ 0 & 0 \end{pmatrix}.$$

We also note that the matrix T_m transforms $\widetilde{w}$ into e_{k+1}, that is, $T_m\widetilde{w} = e_{k+1}$ and for $j = k+1, \ldots, m$, we have $T_m e_{j-1} = e_j$. Fig. 2 shows the structures of the matrices R_m , T_m and $\widetilde{R}_m$ at the end of a SGMRES-IR cycle.

In an actual implementation, we point out how the computation of the matrix $\widetilde{R}_m$ is carried out. First partition R_m in the form

$$R_m = \begin{pmatrix} R_k & \widehat{R}_k \\ 0 & R_{m-k} \end{pmatrix},$$

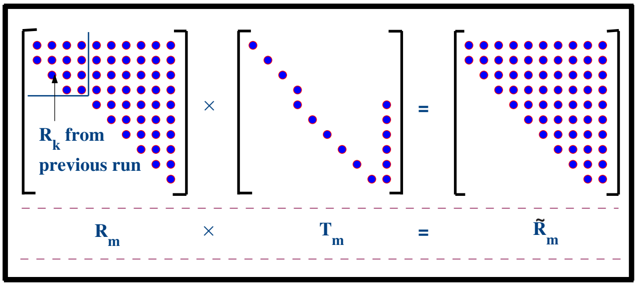


Fig. 2. Structure of $R_m T_m = \widetilde{R}_m$ after a run involving deflation using implicit restarting

where the matrix R_k is upper triangular, the matrix $\widehat{R}_k$ of order $k \times (m-k)$ is full and the matrix R_{m-k} of order $m-k$ is upper triangular. Then

$$\widetilde{R}_m = R_m T_m = \begin{pmatrix} R_k & \widehat{R}_k \\ 0 & R_{m-k} \end{pmatrix} \begin{pmatrix} I_k & 0 \\ \widetilde{T} & T_{m-k} \end{pmatrix} = \begin{pmatrix} R_k & \widehat{R}_k T_{m-k} \\ R_{m-k}\widetilde{T} & R_{m-k} T_{m-k} \end{pmatrix}.$$

Forming the matrix $\widehat{R}_k T_{m-k}$ requires the computation of only the last column in this matrix and this can be done using $mk - k^2$ multiplications. The matrix $R_{m-k}\widetilde{T}$ has only one non-zero entry in the $(k+1,\, m-k)$ position and finally we see that forming the $R_{m-k}T_{m-k}$ only requires the computation of $m-k$ scalar entries in the last column. This gives a total of $(m-k)(k+1)+1$ scalar multiplications for forming $\widetilde{R}_m$. Finally, once the matrix $\widetilde{R}_m$ is formed, we can obtain the shifts associated with the implicit restarting by solving an eigenvalue problem similar to (7).

The SGMRES-IR algorithm is described next.

Algorithm 2. SGMRES-IR

Input: A, b, parameters m, k, tolerance ϵ_s, initial approximation $x_0 := 0$.
Output: Approximate solution x_m and residual r_m.

```
1. Start: $r_0 := b$, $\rho_0 := \|r_0\|_2$, $r_0 := r_0/\rho_0$, $\rho := 1$, $p = m-k$ and $w_1 := Ar_0/\|Ar_0\|_2$;
2. First cycle: Apply Algorithm 3.1 to produce $Y_m$, $W_m$, $R_m$, $r_m$, $x_m$, $\widetilde{w}$;
   Let $r_0 := r_m$ and $x_0 := x_m$;
   if $\|r_m\|_2/\|b\|_2 \le \epsilon_s$ is satisfied then stop, else $\rho := 1$, $\rho_0 := \|r_0\|_2$ and $r_0 := r_0/\rho_0$; endif
3. Eigenvalue problem: Compute $T_m$ and $\widetilde{R}_m := R_m T_m$. Solve $\widetilde{R}_m \widehat{g}_j = \tilde{\theta}_j \widehat{g}_j$, for appropriate $\widehat{g}_j$.
4. Implicit QR Steps: Perform implicit QR with $m-k$ largest eigenvalues.
5. Simpler Arnoldi extension: Apply Algorithm 5.1 with $W_k := W_k^{\mathrm{new}}$, $Y_k := Y_k^{\mathrm{new}}$, $R_k := R_k^{\mathrm{new}}$, $r_0$, $A$, $k$ and $m$ as inputs to generate $Y_m$, $W_m$, $R_m$.
   Compute $\xi_i := w_i^{\mathrm{T}} r_{i-1}$ and $r_i := r_{i-1} - \xi_i w_i$ where $r_k$ is $r_0$ for $i = k+1,\, k+2,\, \ldots,\, m$.
   Update $\rho$ using $\rho := \rho \sin(\cos^{-1}(\xi_i)/\rho)$ and if $\rho/\|b\|_2 \le \epsilon_s$, we go to 7 after the $i^{th}$ step.
6. Form the approximate solution: Solve $R_m \widehat{y} = \widetilde{w}$ as shown in section 3.
   $Y_m := [W_{k-1},\, y_k,\, r_0,\, w_{k+1}, \ldots, w_{m-1}]$; $x_m = x_0 + \rho_0 Y \widehat{y}$;
7. Restart: if $\|r_m\|_2/\|b\|_2 \le \epsilon_s$ then stop, else $x_0 := x_m$; $\rho_0 := \|r_m\|_2$; $r_0 := r_m/\rho_0$; endif
   Set $\rho := 1$ and go to 3.
```

4 SGMRES-IR Subspace

For implicitly restarted GMRES, Morgan [10] showed that implicit restarting generates the subspace

$$S_{m,\,k} = \operatorname{span}\left\{r_0,\, Ar_0,\, \ldots,\, A^{m-k-1}r_0,\, \widetilde{u}_1,\, \ldots,\, \widetilde{u}_k\right\}, \tag{19}$$

where the vectors $\widetilde{u}_j$ for $j = 1, 2, \ldots, k$ are harmonic Ritz vectors.

Similar to the proof given by Morgan that implicit restarting of GMRES generates subspace (19), we give here an analoguous proof in which indicates that Simpler GMRES with implicit restarting generates the same subspace. The difference in the method of the proof is that we use relationships that hold for the SGMRES algorithm.

For $j = 1, 2, \ldots, m$, we let $\widetilde{u}_j = Y_m T_m \widehat{g}_j = Y_m \widetilde{g}_j$ denote the harmonic Ritz vector corresponding to the harmonic Ritz value $\tilde{\theta}_j$. The SGMRES residual vector r_m belongs to $\mathcal{K}_{m+1}(A; r_0)$ and is orthogonal to span$\{W_m\}$ and has roots at the harmonic Ritz values. We can thus write

$$r_m = \alpha \prod_{l=1}^{m} (A - \tilde{\theta}_l I) r_0.$$

The harmonic residual vector $\widetilde{r}_j = A\widetilde{u}_j - \tilde{\theta}_j \widetilde{u}_j$ also belongs to $\mathcal{K}_{m+1}(A; r_0)$ and is orthogonal to span$\{W_m\}$. We thus have, for each j,

$$(A - \tilde{\theta}_j I)\widetilde{u}_j = \alpha_j \prod_{l=1}^{m} (A - \tilde{\theta}_l I) r_0.$$

This means that

$$\widetilde{u}_j = \phi_j(A) r_0, \tag{20}$$

where ϕ_j is the polynomial such that

$$\phi_j(\zeta) = \alpha_j \prod_{l=1,\, l\neq j}^{m} \left(\zeta - \tilde{\theta}_l\right), \tag{21}$$

and α_j is a scalar.
The following result proved by induction in Goossens and Roose [4] is useful for relating polynomials in A to polynomials in $\widetilde{R}_m$.

Lemma 2. *For $j = 1, 2, \ldots, m$, we have*

$$A^j r_0 = W_m \widetilde{R}_m^j \widetilde{w}.$$

We can then prove the following result.

Theorem 1. *For the polynomial ϕ_j in (20), the eigenvector $\widehat{g}_j$ corresponding to the eigenvalue $\tilde{\theta}_j$ of $\widetilde{R}_m$ satisfies*

$$\widehat{g}_j = \phi_j(\widetilde{R}_m)\widetilde{w}. \tag{22}$$

Proof. Since $\phi_j(\zeta)$ has strict degree $m-1$, we can write

$$\phi_j(\zeta) = \gamma_{m-1}\zeta^{m-1} + \cdots + \gamma_1 \zeta + \gamma_0.$$

Then

$$\begin{aligned}\widetilde{u}_j &= \phi_j(A)r_0 \\ &= \gamma_{m-1}A^{m-1}r_0 + \cdots + \gamma_1 Ar_0 + \gamma_0 r_0 \\ &= W_m\left(\gamma_{m-1}\widetilde{R}_m^{m-1}\widetilde{w} + \cdots + \gamma_1\widetilde{R}_m\widetilde{w}\right) + \gamma_0 r_0.\end{aligned}$$

Since $W_m^{\mathrm{T}}Y_m = T_m^{-1}$, we find that

$$W_m^{\mathrm{T}}\widetilde{u}_j = W_m^{\mathrm{T}}Y_m T_m\widehat{g}_j = \widehat{g}_j.$$

Then

$$\begin{aligned}\widehat{g}_j &= W_m^{\mathrm{T}}\left(W_m\left(\gamma_{m-1}\widetilde{R}_m^{m-1}\widetilde{w} + \cdots + \gamma_1\widetilde{R}_m\widetilde{w}\right) + \gamma_0 r_0\right) \\ &= \gamma_{m-1}\widetilde{R}_m^{m-1}\widetilde{w} + \cdots + \gamma_1\widetilde{R}_m\widetilde{w} + \gamma_0\widetilde{w} \\ &= \phi_j(\widetilde{R}_m)\widetilde{w}.\end{aligned}$$

□

The following result is similar to Lemma 5.11 given by Morgan [10] for GMRES-IR.

Lemma 3. *For $j \leq m$, there exists scalars γ and β_1, β_2, …, β_j such that*

$$e_j = \gamma\prod_{l=1}^{j}\left(\widetilde{R}_m - \beta_l I\right)\widetilde{w}. \tag{23}$$

Proof. Since the jth column w_j of W_m belongs to $\mathcal{K}_{j+1}(A;\, r_0)$, we have

$$w_j = \gamma_j A^j r_0 + \cdots + \gamma_1 Ar_0 + \gamma_0 r_0.$$

Then

$$e_j = W_m^{\mathrm{T}}w_j = \left(\gamma_j\widetilde{R}_m^j + \cdots + \gamma_1\widetilde{R}_m + \gamma_0\right)\widetilde{w}.$$

It thus follows that there exists scalars γ and β_1, β_2, …, β_j such that

$$e_j = \gamma\prod_{l=1}^{j}\left(\widetilde{R}_m - \beta_l I\right)\widetilde{w}.$$

□

Let ψ denote the polynomial of degree $m-k$ given by

$$\psi(\zeta) = \prod_{l=k+1}^{m}\left(\zeta - \tilde{\theta}_l\right). \tag{24}$$

Then the following result is obtained.

Lemma 4. *For $t \in span\{e_1, e_2, \ldots, e_k\}$,*

$$\psi(\widetilde{R}_m)t \in span\{\widehat{g}_1, \ldots, \widehat{g}_k\}.$$

Proof. For $j \leq k$,

$$\begin{aligned}\psi(\widetilde{R}_m)e_j &= \gamma\psi(\widetilde{R}_m)\prod_{l=1}^{j}\left(\widetilde{R}_m - \beta_l I\right)\widetilde{w}\\ &= \gamma\prod_{l=1}^{j}\left(\widetilde{R}_m - \beta_l I\right)\psi(\widetilde{R}_m)\widetilde{w}.\end{aligned}$$

Using (21) and Theorem 1,

$$\begin{aligned}\widehat{g}_j &= \phi_j(\widetilde{R}_m)\widetilde{w}\\ &= \alpha_j \prod_{l=1,\, l\neq j}^{m}\left(\widetilde{R}_m - \tilde{\theta}_l I\right)\widetilde{w}\\ &= \alpha_j\left(\prod_{l=1,\, l\neq j}^{k}\left(\widetilde{R}_m - \tilde{\theta}_l I\right)\right)\psi(\widetilde{R}_m)\widetilde{w}.\end{aligned}$$

Since the $\tilde{\theta}_j$'s are distinct, it is possible to express the polynomials $\prod_{l=1}^{j}(\zeta - \beta_l)$ as a combination of the polynomials $\alpha_j\prod_{l=1,\, l\neq j}^{k}\left(\zeta - \tilde{\theta}_l\right)$. Therefore $\psi(\widetilde{R}_m)e_j \in$ span$\{\widehat{g}_1, \ldots, \widehat{g}_k\}$.

□

The following result can be proved using similar arguments as in Morgan [10, Lemma 5.12].

Lemma 5. *Let the matrix Q and shifts σ_i be from the QR iteration. Then for $j \leq k$,*

$$Qe_j = \prod_{l=1}^{m-k}\left(\widetilde{R}_m - \sigma_l I\right)t,$$

where $t \in span\{e_1, \ldots, e_k\}$.

Theorem 2. *The SGMRES-IR method generates the subspace*

$$span\{r_0, Ar_0, \ldots, A^{m-k-1}r_0, \widetilde{u}_1, \widetilde{u}_2, \ldots, \widetilde{u}_k\}. \tag{25}$$

Proof. The unwanted harmonic Ritz values, $\tilde{\theta}_{k+1}, \ldots, \tilde{\theta}_m$ are used as shifts for the QR iteration in Lemma 5. For ψ defined in (24), we thus have $\prod_{l=1}^{m-k}\left(\widetilde{R}_m - \tilde{\theta}_{k+l}I\right) = \psi(\widetilde{R}_m)$. Using Lemma 5, we have $Qe_j = \psi(\widetilde{R}_m)t$ where $t \in$ span$\{e_1, \ldots, e_k\}$. Applying Lemma 4, we have

$$Qe_j \in \text{span}\{\widehat{g}_1, \ldots, \widehat{g}_k\}. \tag{26}$$

After $m - k$ steps of the implicit QR algorithm, the first k columns $\{y_1, y_2, \ldots, y_k\}$ of $Y_m^{(m-k)}$ are such that $y_j = Y_m T_m Q e_j$. From (26), we deduce that

$$\begin{aligned} \operatorname{span}\{y_1, y_2, \ldots, y_k\} &= \operatorname{span}\{Y_m \widetilde{g}_1, \ldots, Y_m \widetilde{g}_k\} \\ &= \operatorname{span}\{\widetilde{u}_1, \ldots, \widetilde{u}_k\}. \end{aligned}$$

Noting that at the time of restart the new r_0 is the normalized residual vector $r_m/\|r_m\|_2$ from the previous run and that

$$A\widetilde{u}_j - \tilde{\theta}_j \widetilde{u}_j = \gamma_j r_0,$$

it follows that $A\widetilde{u}_j$ is a combination of r_0 and $\widetilde{u}_j$. Applying the Simpler Arnoldi extension algorithm shows that the basis vector w_{k+1} is a combination of r_0, Ar_0 and $\widetilde{u}_1, \ldots, \widetilde{u}_k$. Iterating the Simpler Arnoldi extension up to step m generates the Krylov subspace (25). □

5 Numerical Experiments

GMRES and Simpler GMRES are mathematically equivalent in the sense that they generate the same sequence of iterates $\{x_i\}_{i\geq 1}$ in exact arithmetic. However, the Simpler GMRES method uses a non-orthogonal basis that may become ill-conditioned. Walker and Zhou [17] showed that the condition number of the Simpler GMRES basis can become large only if the relative residual norm is small and as pointed out by Rozloznik and Strakos [14], this drawback may not be a serious one unless a very accurate approximation to the solution is sought.

In this section, we compare the performances of the two variants of augmented GMRES for solving different linear systems. The iterations are stopped using the criterion $\|r_i\|_2/\|r_0\|_2 \leq 10^{-10}$. We have chosen the initial approximation x_0 to be a vectors of zeros. In the tables, we provide information on the number of matrix-vector products (matvecs) required for convergence, the relative residual norm $\|r_i\|_2/\|r_0\|_2$, the actually computed residual norm $\|b - Ax_i\|_2/\|b\|_2$ and the computational time in seconds.

Example 1: We first consider matrix SAYLR4 from the Harwell-Boeing Sparse matrix Collection. This matrix has dimension 3564 with an average of 5.3 nonzeros per row. In this experiment, the maximum size of the Krylov subspace is $m = 10$, the number of approximate eigenvectors used is $k = 2$ or $k = 4$ and the ILU(0) preconditioning was used. The numerical results are shown in Table 1. The numerical results indicate that both the GMRES and Simpler GMRES are successful in reducing the number of matrix-vector products required for convergence and that both algorithms maintain comparable accuracy. Also, the Simpler variants are much more efficient in terms of computational time required for convergence.

Table 1. Numerical Results for matrix SAYLR4

Method	matvecs	$\|r_i\|_2/\|r_0\|_2$	$\|b - Ax_i\|_2/\|b\|_2$	time(s)
GMRES(10)	837	9.59×10^{-11}	7.11×10^{-10}	6.6
SGMRES(10)	837	9.51×10^{-11}	7.28×10^{-10}	1.3
GMRES-IR(10, 2)	92	9.93×10^{-11}	6.86×10^{-10}	1.5
SGMRES-IR(10, 2)	92	9.93×10^{-11}	7.18×10^{-10}	0.6
GMRES-IR(10, 4)	81	6.44×10^{-11}	5.70×10^{-10}	1.0
SGMRES-IR(10, 4)	81	6.44×10^{-11}	6.12×10^{-10}	0.5

Table 2. Numerical Results for matrix SHERMAN4

Method	matvecs	$\|r_i\|_2/\|r_0\|_2$	$\|b - Ax_i\|_2/\|b\|_2$	time(s)
GMRES(25)	855	9.85×10^{-11}	9.85×10^{-11}	2.9
SGMRES(25)	855	9.85×10^{-11}	9.85×10^{-11}	0.5
GMRES-IR(25, 4)	193	9.56×10^{-11}	9.56×10^{-11}	1.2
SGMRES-IR(25, 4)	193	9.56×10^{-11}	9.67×10^{-11}	0.7
GMRES-IR(25, 8)	169	9.37×10^{-11}	9.37×10^{-11}	1.1
SGMRES-IR(25, 8)	169	9.37×10^{-11}	9.67×10^{-11}	0.6

Table 3. Results when $m = 20$ and $k = 3$

Method	q	matvecs	$\|r_i\|_2/\|r_0\|_2$	$\|b - Ax_i\|_2/\|b\|_2$
GMRES	10^5	658	1×10^{-10}	1×10^{-10}
SGMRES	10^5	658	1×10^{-10}	6.96×10^{-9}
GMRES-IR	10^5	> 280	7.28×10^{-11}	3.41×10^{-2}
SGMRES-IR	10^5	> 280	9.92×10^{-11}	4.25×10^{-2}
GMRES	10^9	659	9.80×10^{-11}	9.83×10^{-11}
SGMRES	10^9	659	9.80×10^{-11}	7.16×10^{-9}
GMRES-IR	10^9	> 500	2.38×10^{-2}	2.38×10^{-2}
SGMRES-IR	10^9	> 500	9.97×10^{-11}	1.17×10^{-2}

Example 2: Our next test problem is SHERMAN4, a nonsymmetric matrix of dimension 1104 and with 3786 nonzero entries. We give results for varying number k of approximate harmonic Ritz vectors used to carry out the deflation process with a maximum Krylov subspace size of 25 without the use of a preconditioner. Table 2 indicates that both the implicitly restarted variants perform better than the GMRES and SGMRES methods. Similar conclusions as those observed for the matrix SAYLR4 hold for the SHERMAN4.

Example 3: We choose a test problem from Morgan [11, Example 4]. The matrix A has diagonal entries given by 1, 2, 3, 4, . . ., 999 and q and the right hand side has all 1's. Morgan used this example to show stability problems with GMRES-IR resulting from the instability of the QR algorithm which is considered to

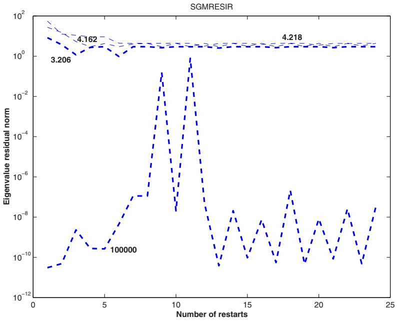


Fig. 3. SGMRES-IR(20, 3) eigenvalue residuals when $q = 10^5$

be numerically backward stable. In fact, this may be related to the forward instability as explained by Parlett and Le [12].

We run GMRES-IR and SGMRES-IR with parameters $m = 20$ and $k = 3$. Table 3 shows the results when $q = 10^5$ and $q = 10^9$. The true residual norms for the implicitly restarted methods stagnate for both case problems. However, for this test problem, we observe that the relative residual norm $\|\check{r}_m\|_2/\|b\|_2$ for the SGMRES-IR iterations decreases erroneously. To explain the stagnation,, we show in Figure 3, the convergence history of some of the harmonic Ritz values generated by SGMRES-IR(20, 3). We observe that SGMRES-IR is not able to develop good approximations to these smallest eigenvalues and thus the method stalls.

6 Conclusion

We have proposed a Simpler GMRES algorithm with implicit restarting for solving linear systems with small eigenvalues. We showed that this new method generates the same subspace as the implicitly restarted GMRES algorithm and involves a transformation matrix that simplifies the computations, thus leading to cheaper algorithms. Numerical tests carried out indicate that in cases where both variants of augmented GMRES are successful, the Simpler GMRES variant maintains comparable accuracy as the GMRES variant and outperforms it in terms of computational time required for convergence. For computational costs concerning the implementation of augmented Simpler GMRES and results indicating cheaper implementations, we refer the reader to the work in [1].

References

1. Boojhawon, R., Bhuruth, M.: Restarted Simpler GMRES augmented with harmonic Ritz vectors. Future Generation Computer Systems 20, 389–397 (2004)
2. Boojhawon, R., Bhuruth, M.: Implementing GMRES with Deflated Restarting via the shifted Arnoldi Process. In: Proceedings of the 2006 Conference on Computational and Mathematical Methods on Science and Engineering, pp. 133–152 (2006)

3. Drkosova, J., Greenbaum, A., Rozloznik, M., Strakos, Z.: Numerical stability of GMRES. BIT 35, 309–330 (1995)
4. Goossens, S., Roose, D.: Ritz and harmonic Ritz values and the convergence of FOM and GMRES. Numer. Lin. Alg. Appl. 6, 281–293 (1999)
5. Greenbaum, A., Rozloznik, M., Strakos, Z.: Numerical behaviour of modified Gram-Schmidt GMRES implementation. BIT 37, 709–719 (1997)
6. Higham, N.: Accuracy and Stability of Numerical Algorithms, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2002)
7. Jiranek, P., Rozloznik, M., Gutknecht, M.: How to make Simpler GMRES and GCR more Stable. SIAM J. Matrix Anal. Appl. 30(4), 1483–1499 (2008)
8. Liesen, J., Rozloznik, M., Strakos, Z.: Least squares residuals and minimal residual methods. SIAM J. Sci. Comput. 23, 1503–1525 (2002)
9. Morgan, R.: A restarted GMRES method augmented with eigenvectors. SIAM J. Matrix Anal. Appl. 16(4), 1154–1171 (1995)
10. Morgan, R.: Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations. SIAM J. Matrix Anal. Appl. 21(4), 1112–1135 (2000)
11. Morgan, R.B.: GMRES with deflated restarting. SIAM J. Sci. Comp. 24, 20–37 (2002)
12. Parlett, B.N., Le, J.: Forward instability of tridiagonal QR. SIAM J. Matrix Anal. Appl. 14, 279–316 (1993)
13. Rozloznik, M.: Numerical Stability of the GMRES Method. PhD thesis, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague (1996)
14. Rozloznik, M., Strakos, Z.: Variants of the residual minimizing Krylov space methods. In: Proceedings of the XI th Summer School Software and Algorithms of Numerical Mathematics, pp. 208–225 (1995)
15. Saad, Y., Schultz, M.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. 7, 865–869 (1986)
16. Sorensen, D.C.: Implicit application of polynomial filters in a k-step Arnoldi method. SIAM J. Matrix Anal. Appl. 13, 357–385 (1992)
17. Walker, H., Zhou, L.: A Simpler GMRES. Numer. Lin. Alg. Appl. 1, 571–581 (1992)
18. Wu, K., Simon, H.: Thick-restart Lanczos method for large symmetric eigenvalue problems. SIAM J. Matrix Anal. Appl. 22, 602–616 (2000)

On the Linearity of Cryptographic Sequence Generators*

Amparo Fuster-Sabater[1], Oscar Delgado-Mohatar[1], and Ljiljana Brankovic[2]

[1] Institute of Applied Physics, C.S.I.C., Serrano 144, 28006 Madrid, Spain
amparo@iec.csic.es, oscar.delgado@iec.csic.es

[2] School of Electrical Engineering and Computer Science, University of Newcastle,
Callaghan, Australia 2308
Ljiljana.Brankovic@newcastle.edu.au

Abstract. In this paper we show that the output sequences of the generalized self-shrinking generator are particular solutions of a binary homogeneous linear difference equation. In fact, all these sequences are just linear combinations of primary sequences weighted by binary coefficients. We show that in addition to the output sequences of the generalized self-shrinking generator, the complete class of solutions of the corresponding binary homogeneous linear difference equation also includes other balanced sequences that are very suitable for cryptographic applications, as they have the same period and even greater linear complexity than the generalized self-shrinking sequences. Cryptographic parameters of all the above mentioned sequences can be analyzed in terms of linear equation solutions.

Keywords: binary sequence, linear difference equation, generalized self-shrinking generator, cryptography.

1 Introduction

Stream ciphers have extensive applications in secure communications, e.g., wireless systems, due to their practical advantages such as easy implementation, high speed and good reliability. From a short secret key, a stream cipher procedure generates a long sequence of pseudorandom bits, commonly referred to as the *keystream sequence*. Most keystream generators are based on maximal-length Linear Feedback Shift Registers (LFSRs) [5] whose output sequences, the so-called m-sequences, are combined in a non linear way in order to produce pseudorandom sequences of cryptographic application. Combinational generators, non-linear filters, clock-controlled generators, irregularly decimated generators, etc, are just some of the most popular keystream sequence generators [9], [13].

* This work was supported in part by CDTI (Spain) and the companies INDRA, Unión Fenosa, Tecnobit, Visual Tool, Brainstorm, SAC and Technosafe under Project Cenit-HESPERIA; by Ministry of Science and Innovation and European FEDER Fund under Project TIN2008-02236/TSI; by The University of Newcastle, CEF G0189479.

D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 586–596, 2010.

Coppersmith, Krawczyk and Mansour [1] proposed the *shrinking generator*. Subsequently, Meier and Staffelbach [12] proposed the *self-shrinking generator*. Finally, Hu and Xiao [8] defined the *generalized self-shrinking generator* that generates a family of pseudorandom sequences suitable for cryptographic applications due to their long periods, good correlation features, excellent run distribution, balancedness, simplicity of implementation, etc. The generalized self-shrinking generator can be seen as a specialization of the shrinking generator as well as a generalization of the self-shrinking generator. In fact, the output sequence of the self-shrinking generator is just an element of the family of generalized self-shrinking sequences. The main idea of all the generators of this kind is the irregular decimation of an m-sequence according to the bits of another one. The decimation result is the output sequence that will be used as keystream sequence. Some cryptanalysis on these decimation generators can be found in ([3], [4], [7], [14], [15], [16]).

The main contributions of this paper are as follows.

1. We show that the generalized self-shrinking sequences are particular solutions of a type of binary coefficient homogeneous linear difference equations.
2. We show that some other solution sequences not included in the previous family also exhibit good properties for their application in cryptography. In general, the solutions of linear different equations provide one with an easy method of generating keystream sequences.

The rest of the paper is organized as follows. In Section 2 we give a definition of the generalized self-shrinking generator and illustrate it with an example, while in Section 3 we take a closer look at linear difference equations with binary coefficients. In Section 4 we present the main result of this paper, that is, we show that is, we show that the generalized self-shrinking sequences are particular solutions of the equations discussed in Section 3, and we analyze the characteristics of other solutions of such equations. We provide a comprehensive example in Section 5 and some concluding remarks in Section 6.

2 The Generalized Self-shrinking Generator

The generalized self-shrinking generator is a binary sequence generator that can be described as follows.

1. It makes use of two sequences: an m-sequence $\{a_n\}$ and a shifted version of such a sequence denoted by $\{v_n\}$.
2. It relates both sequences by means of a simple decimation rule to generate the output sequence.

The result of the above steps is a family of generalized self-shrinking sequences that can formally be defined as follows [8].

Definition 1. *Let $\{a_n\}$ be an m-sequence over $GF(2)$ with period $2^L - 1$ generated from a LFSR of primitive characteristic polynomial of degree L. Let G be an L-dimensional binary vector defined as:*

$$G = (g_0, g_1, g_2, \ldots, g_{L-1}) \in GF(2)^L. \tag{1}$$

The n-th element of the sequence $\{v_n\}$ *is defined as:*

$$v_n = g_0 a_n \oplus g_1 a_{n-1} \oplus g_2 a_{n-2} \oplus \ldots \oplus g_{L-1} a_{n-L+1}. \tag{2}$$

where the sub-indexes of the sequence $\{a_n\}$ *are reduced mod* $2^L - 1$ *and the symbol* $\oplus$ *represents the XOR logic operation. For* $n \geq 0$ *the decimation rule is very simple:*

1. *If* $a_n = 1$*, then* v_n *is output.*
2. *If* $a_n = 0$*, then* v_n *is discarded and there is no output bit.*

In this way, an output sequence $b_0\, b_1\, b_2 \ldots$ *denoted by* $\{b_n\}$ *or* $\{b(G)\}$ *is generated. Such a sequence is called the generalized self-shrinking sequence associated with* G*.*

Note that the sequence $\{v_n\}$ is nothing but a shifted version of the sequence $\{a_n\}$. When G ranges over $GF(2)^L - (0, \ldots, 0)$, then $\{v_n\}$ corresponds to the $2^L - 1$ possible shifts of $\{a_n\}$. In addition, the set of sequences denoted by $B(a) = \{\{b(G)\}, G \in GF(2)^L\}$ is the family of generalized self-shrinking sequences based on the m-sequence $\{a_n\}$.

Example. [8] For the 4-degree m-sequence $\{a_n\} = \{111101011001000\}$ whose characteristic polynomial is $x^4 + x^3 + 1$, we get 16 generalized self-shrinking sequences based on the sequence $\{a_n\}$ (see [8]):

1. $G = (0000), \{b(G)\} = 00000000 \sim$
2. $G = (1000), \{b(G)\} = 11111111 \sim$
3. $G = (0100), \{b(G)\} = 01110010 \sim$
4. $G = (1100), \{b(G)\} = 10001101 \sim$
5. $G = (0010), \{b(G)\} = 00111100 \sim$
6. $G = (1010), \{b(G)\} = 11000011 \sim$
7. $G = (0110), \{b(G)\} = 01001110 \sim$
8. $G = (1110), \{b(G)\} = 10110001 \sim$
9. $G = (0001), \{b(G)\} = 00011011 \sim$
10. $G = (1001), \{b(G)\} = 11100100 \sim$
11. $G = (0101), \{b(G)\} = 01101001 \sim$
12. $G = (1101), \{b(G)\} = 10010110 \sim$
13. $G = (0011), \{b(G)\} = 00100111 \sim$
14. $G = (1011), \{b(G)\} = 11011000 \sim$
15. $G = (0111), \{b(G)\} = 01010101 \sim$
16. $G = (1111), \{b(G)\} = 10101010 \sim$

It is important to note that the generated sequences are not 16 different sequences. In fact, sequences 5 and 6 are shifted versions of the same sequence and the same is true for sequences 11 and 12 and sequences 15 and 16. At the same time, sequences 3, 7, 10 and 13 correspond to a unique sequence, as do sequences 4, 8, 9 and 14. Periods, linear complexities and number of different sequences obtained from this generator will be studied in the following sections in terms of solutions of linear difference equations.

3 Linear Difference Equations with Binary Coefficients

Throughout this paper, we consider the following type of linear difference equation with binary coefficients:

$$(E^r \oplus \sum_{j=1}^{r} c_j \; E^{r-j}) \; z_n = 0, \qquad n \geq 0 \tag{3}$$

where $z_n \in GF(2)$ is the n-th term of a binary sequence $\{z_n\}$ that satisfies the above equation. We use symbol E to denote a shifting operator that operates on the terms z_n of a solution sequence, i.e. $E^j z_n = z_{n+j}$. The coefficients c_j are constant binary coefficients $c_j \in GF(2)$, r is an integer and the symbol $\oplus$ represents the XOR logic operation. The r-degree characteristic polynomial of (3) is:

$$P(x) = x^r + \sum_{j=1}^{r} c_j \; x^{r-j}, \tag{4}$$

and specifies the linear recurrence relationship of the sequence $\{z_n\}$. This means that its n-th term, z_n, can be written as a linear combination of the previous r terms:

$$z_n \oplus \sum_{j=1}^{r} c_j \; z_{n-j} = 0, \qquad n \geq r. \tag{5}$$

If $P(x)$ is a primitive polynomial [10] and α is one of its roots, then

$$\alpha, \; \alpha^2, \; \alpha^{2^2}, \ldots, \; \alpha^{2^{(r-1)}} \in GF(2)^r \tag{6}$$

are the r different roots of such a polynomial (see [11]). In this case, it can be proved [10] that the solution of (3) is a sequence of the form:

$$z_n = \sum_{j=0}^{r-1} A^{2^j} \; \alpha^{2^j n}, \qquad n \geq 0 \tag{7}$$

where A is an arbitrary element in $GF(2)^r$. That is, $\{z_n\}$ is an m-sequence [6] of characteristic polynomial $P(x)$ and period $2^r - 1$ whose starting point is determined by the value of A. If $A = 0$, then the solution of (3) is the identically null sequence.

Let us generalize the above difference equations to a more complex type of linear difference equations whose roots have a multiplicity greater than 1. In fact, we are going to consider equations of the form:

$$(E^r \oplus \sum_{j=1}^{r} c_j \; E^{r-j})^p \; z_n = 0, \qquad n \geq 0 \tag{8}$$

p being an integer $p > 1$. The characteristic polynomial $P_M(x)$ of this type of equation is:

$$P_M(x) = P(x)^p = (x^r + \sum_{j=1}^{r} c_j \; x^{r-j})^p. \tag{9}$$

In this case, the roots of $P_M(x)$ are the same as those of the polynomial $P(x)$, that is, $(\alpha, \alpha^2, \alpha^{2^2}, \ldots, \ \alpha^{2^{(r-1)}})$, but with multiplicity p. Now the solutions of (8) are [2]:

$$z_n = \sum_{i=0}^{p-1} (\binom{n}{i} \sum_{j=0}^{r-1} A_i^{2^j} \ \alpha^{2^j n}), \qquad n \geq 0 \tag{10}$$

where $A_i \in GF(2)^r$ and the $\binom{n}{i}$ are binomial coefficients modulo 2.

In brief, the n-th term of a solution sequence $\{z_n\}$ is the bit-wise XOR logic operation of the n-th term of p sequences $\{\sum\limits_{j=0}^{r-1} \ A_i^{2^j} \ \alpha^{2^j n}\}$ $(0 \leq i < p)$ weighted by binomial coefficients.

In fact, when n takes successive values each binomial coefficient $\binom{n}{i}$ $(n \geq i \geq 0)$ defines a *primary sequence* with constant period T_i. In Table 1, the first binomial coefficients with their corresponding primary sequences and periods are depicted. From Table 1, it is easy to see that the generation of such sequences follows a simple general rule. Indeed, the 2^m primary sequences associated with $\binom{n}{i}$ for $(2^m \leq i < 2^{m+1})$ (m being a non-negative integer) have period $T_i = 2^{m+1}$ and their digits are:

1. The first 2^m bits are 0's.
2. The other bits are the first 2^m bits of each one of the previous 2^m primary sequences, respectively.

Let us consider a simple example. According to Table 1 and for $m = 2$, we have 2^2 primary sequences S_i with $(2^2 \leq i < 2^3)$. The sequence S_4 has 2^2 0's and the 2^2 first digits of S_0. In the same way, the sequence S_5 has 2^2 0's and the 2^2 first digits of S_1. The sequence S_6 has 2^2 0's and the 2^2 first digits of S_2 while the sequence S_7 has 2^2 0's and the 2^2 first digits of S_3. In general, the digits of S_j are related with those of S_i by means of the expression $i = 2^m + j$. Therefore, generation and handling of such sequences is very easy.

4 Main Results

We now present the main results concerning generalized self-shrinking sequences and linear difference equations.

Theorem 1. *The family of generalized self-shrinking sequences B(a) based on the m-sequence* $\{a_n\}$ *are particular solutions of the homogeneous linear difference equation:*

$$(E \oplus 1)^p \ z_n = 0, \qquad p = 2^{L-1} \ , \tag{11}$$

whose characteristic polynomial is $(x + 1)^p$.

Proof. According to [8], the periods of the generalized self-shrinking sequences $B(a)$ are $T \in \{1, 2, 2^{L-1}\}$ where L is the degree of the primitive characteristic polynomial of the m-sequence $\{a_n\}$. Thus, the period T of any generalized self-shrinking sequence divides 2^{L-1}, i.e., it is a power of 2. Hence over $GF(2)$,

Table 1. Binomial coefficients, primary sequences and periods T_i

Binomial coeff.	Primary sequences	T_i
$\binom{n}{0}$	$S_0 = \{1, 1, 1, 1, 1, 1, 1, 1 \sim\}$	$T_0 = 1$
$\binom{n}{1}$	$S_1 = \{0, 1, 0, 1, 0, 1, 0, 1 \sim\}$	$T_1 = 2$
$\binom{n}{2}$	$S_2 = \{0, 0, 1, 1, 0, 0, 1, 1 \sim\}$	$T_2 = 4$
$\binom{n}{3}$	$S_3 = \{0, 0, 0, 1, 0, 0, 0, 1 \sim\}$	$T_3 = 4$
$\binom{n}{4}$	$S_4 = \{0, 0, 0, 0, 1, 1, 1, 1 \sim\}$	$T_4 = 8$
$\binom{n}{5}$	$S_5 = \{0, 0, 0, 0, 0, 1, 0, 1 \sim\}$	$T_5 = 8$
$\binom{n}{6}$	$S_6 = \{0, 0, 0, 0, 0, 0, 1, 1 \sim\}$	$T_6 = 8$
$\binom{n}{7}$	$S_7 = \{0, 0, 0, 0, 0, 0, 0, 1 \sim\}$	$T_7 = 8$

$x^T + 1 = (x + 1)^T$. On the other hand, if $f(x)$ is the characteristic polynomial of the shortest linear recursion satisfied by a generalized self-shrinking sequence, then the condition $f(x)|x^T + 1$ implies that $f(x)$ is of the form:

$$f(x) = (x + 1)^{LC} \tag{12}$$

where LC is its linear complexity. At the same time, it is a well known fact [8] that the linear complexity of a periodic sequence is less than or equal to its least period. Thus, for a generalized self-shrinking sequence $LC \leq 2^{L-1}$ and the polynomial of the shortest linear recursion $f(x)$ divides the characteristic polynomial of (11). Therefore, the generalized self-shrinking sequences satisfied (11) and are particular solutions of such an homogeneous linear difference equation. □

We now analyze in detail the characteristics of the sequences that satisfy the previous linear difference equation. According to (10), the solutions of the difference equation given in (11) are of the form:

$$z_n = \binom{n}{0} A_0 \oplus \binom{n}{1} A_1 \oplus \ldots \oplus \binom{n}{p-1} A_{p-1}, \qquad n \geq 0 \tag{13}$$

where $A_i \in GF(2)$ are binary coefficients, $\alpha = 1$ is the unique root with multiplicity p of the polynomial $(x + 1)$ of degree $r = 1$ and $\binom{n}{i}$ $(0 \leq i < p)$ are the binomial coefficients mod 2. Note that the sequence $\{z_n\}$ is just the bitwise XOR logic operation of primary sequences weighted by the corresponding coefficients A_i. Indeed, different choices of coefficients A_i will give rise to different sequences with different characteristics. In addition, it is important to note that not all the solutions $\{z_n\}$ of (11) are generalized self-shrinking sequences, e.g., there are sequences with periods different from $\{1, 2, 2^{L-1}\}$ that satisfy the difference equation although they are not obtained from the generalized self-shrinking generator. Similarly, there are solution sequences with period 2^{L-1} that have not been generated by the generalized self-shrinking generator. From

(13) particular features of the solution sequences and consequently of the generalized self-shrinking sequences can be easily determined. All of them are related to the choice of the p-tuple $(A_0, A_1, A_2, \ldots, A_{p-1})$ of binary coefficients.

4.1 Periods of the Solution Sequences

According to Section 3, the periods of the primary sequences are just powers of 2. Moreover, according to (13) the sequence $\{z_n\}$ is the bit-wise XOR of sequences with different periods, all of them powers of 2. Thus, the period of a sequence $\{z_n\}$ is the maximum period of the primary sequences involved in (13). In fact, the period of $\{z_n\}$ is the period T_i corresponding to the primary sequence with the greatest index i such that $A_i \neq 0$.

4.2 Linear Complexity of the Solution Sequences

According to [10], the linear complexity of a sequence equals the number and multiplicity of the characteristic polynomial roots that appears in its linear recurrence relationship. Therefore, coming back to (13) and analyzing the coefficients A_i, the linear complexity of $\{z_n\}$ can be computed. In fact, we have a unique root 1 with maximal multiplicity p. Thus, if i is the greatest index $(0 \leq i < p)$ for which $A_i \neq 0$, then the linear complexity LC of the sequence $\{z_n\}$ will be:

$$LC = i + 1 \tag{14}$$

as it will be the multiplicity of the root 1.

4.3 Number of Different Solution Sequences

In order to count the number of different sequences $\{z_n\}$ that are solutions of (11), the choice of the coefficients A_i in (13) must also be considered. If i is the greatest index $(0 \leq i < p)$ for which $A_i \neq 0$, then there are 2^i possible choices of the i-tuple $(A_0, A_1, A_2, , A_{i-1})$ for the sequence $\{z_n\}$ in (13). On the other hand, as the period of such sequences is T_i, the number of different sequences N_i will be:

$$N_i = 2^i / T_i \qquad (0 \leq i < p). \tag{15}$$

The total number N_{total} of different solution sequences of the linear difference equation (11) will be:

$$N_{total} = \sum_{i=0}^{p-1} N_i. \tag{16}$$

In this computation the null sequence corresponding to the null p-tuple is excluded.

In summary, the choice of coefficients A_i allows one to generate binary sequences with controllable periods and linear complexity.

5 An Illustrative Example

Let us now consider the generalized self-shrinking generator introduced in Section 2. In fact, for the 4-degree m-sequence $\{a_n\} = \{111101011001000\}$, the family of generalized self-shrinking sequences $B(a)$ are solutions of the equation:

$$(E \oplus 1)^p \, b_n = 0, \qquad p = 2^3 \; , \tag{17}$$

whose general form is:

$$b_n = \binom{n}{0} A_0 \oplus \binom{n}{1} A_1 \oplus \ldots \oplus \binom{n}{7} A_7, \qquad n \geq 0 \tag{18}$$

Different choices of the 8-tuple $(A_0, A_1, \ldots, A_7)$ can be considered:

1. For $A_i = 0 \;\; \forall \; i$, the solution sequence $\{b_n\} = \{0\}$ is the identically null sequence that corresponds to the generalized self-shrinking sequence:
$$G = (0000), \{b(G)\} = 00000000 \sim .$$
2. For $A_0 \neq 0, \;\; A_i = 0 \;\; \forall \; i > 0$, the solution sequence $\{b_n\} = \{1111 \sim\}$ is the identically 1 sequence that corresponds to the generalized self-shrinking sequence:
$$G = (1000), \{b(G)\} = 11111111 \sim .$$
A sequence with period $T_0 = 1$ and $LC_0 = 1$.
3. For $A_1 \neq 0, \;\; A_i = 0 \;\; \forall \; i > 1$, there is a unique solution sequence $\{b_n\}$ with period $T_1 = 2$ and $LC_1 = 2$. The pair $(A_0 = 0, A_1 = 1)$ generates $\{b_n\} = \{01 \sim\}$ that corresponds to the generalized self-shrinking sequence:
$$G = (0111), \{b(G)\} = 01010101 \sim .$$
The pair $(A_0 = 1, A_1 = 1)$ generates $\{b_n\} = \{10 \sim\}$ that corresponds to the generalized self-shrinking sequence:
$$G = (1111), \{b(G)\} = 10101010 \sim .$$
They are two shifted versions of the same sequence.
4. For $A_2 \neq 0, \;\; A_i = 0 \;\; \forall \; i > 2$, there is a unique and balanced solution sequence $\{b_n\}$ with period $T_2 = 4$ and $LC_2 = 3$. For example, the 3-tuple $(A_0 = 0, A_1 = 0, A_2 = 1)$ generates $\{b_n\} = \{0011 \sim\}$. Other 3-tuples with $A_2 = 1$ give rise to shifted versions of the same sequence. In this case, there is no generalized self-shrinking sequence with such characteristics.
5. For $A_3 \neq 0, \;\; A_i = 0 \;\; \forall \, i > 3$, there are two non balanced different sequences with period $T_3 = 4$ and $LC_3 = 4$. For example, the 4-tuple $(A_0 = 0, A_1 = 1, A_2 = 1, A_3 = 1)$ generates $\{b_n\} = \{0111 \sim\}$ with three 1's, while the 4-tuple $(A_0 = 0, A_1 = 0, A_2 = 0, A_3 = 1)$ generates $\{b_n\} = \{0001 \sim\}$ with only one 1. Other 4-tuples with $A_3 = 1$ give rise to shifted versions of both sequences. In this case, there is no generalized self-shrinking sequence with such characteristics.

6. For $A_4 \neq 0,\ A_i = 0\ \ \forall\ i > 4$, there are two balanced different sequences with period $T_4 = 8$ and $LC_4 = 5$. For example, the 5-tuple $(A_0 = 0, A_1 = 0, A_2 = 1, A_3 = 0, A_4 = 1)$ generates $\{b_n\} = \{00111100 \sim\}$ that corresponds to the generalized self-shrinking sequence:

   $$G = (0010), \{b(G)\} = 00111100 \sim .$$

   Moreover, a shifted version of this sequence $\{b_n\} = \{11000011 \sim\}$ for the 5-tuple (1,0,1,0,1) corresponds to the generalized self-shrinking sequence:

   $$G = (1010), \{b(G)\} = 11000011 \sim .$$

   The 5-tuple $(A_0 = 0, A_1 = 1, A_2 = 1, A_3 = 0, A_4 = 1)$ generates $\{b_n\} = \{01101001 \sim\}$ that corresponds to the generalized self-shrinking sequence:

   $$G = (0101), \{b(G)\} = 01101001 \sim .$$

   The last two sequences are shifted versions of the self-shrinking sequence associated with $\{a_n\}$.
7. For $A_5 \neq 0,\ A_i = 0\ \ \forall\ i > 5$, there are four not all balanced different sequence with period $T_5 = 8$ and $LC_5 = 6$. For example, the 6-tuple $(A_0 = 0, A_1 = 1, A_2 = 1, A_3 = 1, A_4 = 0, A_5 = 1)$ generates $\{b_n\} = \{01110010 \sim\}$ that corresponds to the generalized self-shrinking sequence:

   $$G = (0100), \{b(G)\} = 01110010 \sim .$$

   Moreover, shifted versions of this sequence correspond to the generalized self-shrinking sequence:

   $$\begin{aligned} G &= (0110), \{b(G)\} = 01001110 \sim . \\ G &= (1001), \{b(G)\} = 11100100 \sim . \\ G &= (0011), \{b(G)\} = 00100111 \sim . \end{aligned}$$

   for the 6-tuples (0,1,0,1,1,1), (1,0,0,1,1,1) and (0,0,1,1,0,1), respectively.
8. For $A_6 \neq 0,\ A_i = 0\ \ \forall\ i > 6$, there are eight not all balanced different sequence with period $T_6 = 8$ and $LC_6 = 7$. None of them corresponds to generalized self-shrinking sequences.
   There are four balanced solution sequences $\{b_n\} = \{01010110 \sim\}$, $\{b_n\} = \{10101001 \sim\}$, $\{b_n\} = \{01011100 \sim\}$ and $\{b_n\} = \{10100011 \sim\}$ with the same period, autocorrelation and greater linear complexity than that of the generalized self-shrinking sequences described in steps 6 and 7.
9. For $A_7 \neq 0,\ A_i = 0\ \ \forall\ i > 7$, there are sixteen different and unbalanced sequences with period $T_7 = 8$ and $LC_7 = 8$. None of them correspond to generalized self-shrinking sequences. Nevertheless, it must be noticed that any generalized self-shrinking sequence in steps 6 and 7. becomes a solution sequence of this class just by complementing the last digit, as the primary sequence corresponding to $A_7 = 1$ is $S_7 = \{00000001\}$. For example, the sequence $\{b_n\} = \{00111101 \sim\}$ corresponds to the one-bit complementation of $G = (0010), \{b(G)\} = 00111100 \sim$ or the sequence $\{b_n\} = \{01101000 \sim\}$ corresponds to the one-bit complementation of $G = (0101), \{b(G)\} = 01101001 \sim$ both described in step 6. The same applies for the generalized self-shrinking sequences in step 7.

Note that the complementation of the last bit of generalized self-shrinking sequences with period $2^L - 1$ means that the resulting sequence includes the primary sequence

$$\binom{n}{2^{L-1}-1} \qquad (n \geq 2^{L-1}-1) \tag{19}$$

This implies that the obtained sequence has period $T = 2^{L-1}$, maximum linear complexity $LC = 2^{L-1}$ and quasi-balancedness as the difference between the number of 1's and 0's is just 1. For a cryptographic range $L = 128$, this difference is negligible. Therefore, the selection of coefficients A_i allows one to control period, linear complexity and balancedness of the solutions sequences.

6 Conclusions

In this paper we have shown that generalized self-shrinking sequences and, consequently, self-shrinking sequences are particular solutions of homogeneous linear difference equations with binary coefficients. At the same time, there are other many solution sequences that are not included in the previous class but have the same or even better cryptographic characteristics. Moreover, the choice of the p-tuple $(A_0, A_1, A_2, \ldots, A_{p-1})$ of binary coefficients allows one:

1. to get all the solutions of the above linear difference equation (13), among them there are sequences with application in stream cipher;
2. to obtain sequences with controllable period, linear complexity and balancedness.

It is important to note that, although generalized self-shrinking sequences and self-shrinking sequences are generated from LFSRs by irregular decimation, in practice they are simple solutions of linear equations. This fact establishes a subtle link between irregular decimation and linearity that can be conveniently exploited in the cryptanalysis of such keystream generators. A natural extension of this work is the generalization of this procedure to many other cryptographic sequences, the so-called interleaved sequences [6], as they present very similar structural properties to those of the sequences obtained from irregular decimation generators.

References

1. Coppersmith, D., Krawczyk, H., Mansour, Y.: The Shrinking Generator. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 22–39. Springer, Heidelberg (1994)
2. Dickson, L.E.: Linear Groups with an Exposition of the Galois Field Theory, pp. 3–71. Dover, New York (1958); An updated reprint can be found, `http://www-math.cudenver.edu/~wcherowi/courses/finflds.html`
3. Fúster-Sabater, A., Caballero-Gil, P.: Strategic Attack on the Shrinking Generator. Theoretical Computer Science 409(3), 530–536 (2008)

4. Fúster-Sabater, A., Caballero-Gil, P.: Cryptanalytic Attack on Cryptographic Sequence Generators: The Class of Clock-Controlled Shrinking Generators. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008, Part II. LNCS, vol. 5073, pp. 668–679. Springer, Heidelberg (2008)
5. Golomb, S.W.: Shift Register-Sequences. Aegean Park Press, Laguna Hill (1982)
6. Gong, G.: Theory and Applications of q-ary Interleaved Sequences. IEEE Trans. Information Theory 41(2), 400–411 (1995)
7. Gomulkiewicz, M., Kutylowski, M., Wlaz, P.: Fault Jumping Attacks against Shrinking Generator.In: Dagstuhl Seminar, Proceedings 06111, Complexity of Boolean Functions (2006) http://drops.dagstuhl.de/opus/volltexte/2006/611
8. Hu, Y., Xiao, G.: Generalized Self-Shrinking Generator. IEEE Trans. Inform. Theory 50, 714–719 (2004)
9. Jennings, S.M.: Multiplexed Sequences: Some Properties. In: Beth, T. (ed.) EUROCRYPT 1982. LNCS, vol. 149, Springer, Heidelberg (1983)
10. Key, E.L.: An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators. IEEE Trans. Informat. Theory 22(6), 732–736 (1976)
11. Lidl, R., Niederreiter, H.: Introduction to Finite Fields and Their Applications. Cambridge University Press, Cambridge (1986)
12. Meier, W., Staffelbach, O.: The Self-Shrinking Generator. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 205–214. Springer, Heidelberg (1995)
13. Menezes, A.J., et al.: Handbook of Applied Cryptography. CRC Press, New York (1997)
14. Mihaljevic, M.J.: A Faster Cryptanalysis of the Self-Shrinking Generator. In: Pieprzyk, J.P., Seberry, J. (eds.) ACISP 1996. LNCS, vol. 1172, Springer, Heidelberg (1996)
15. Zenner, E., Krause, M., Lucks, S.: Improved cryptanalysis of the self-shrinking generator. In: Varadharajan, V., Mu, Y. (eds.) ACISP 2001. LNCS, vol. 2119, pp. 21–35. Springer, Heidelberg (2001)
16. Zhang, B., Feng, D.: New Guess-and-Determine Attack on the Self-Shrinking Generator. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 54–68. Springer, Heidelberg (2006)

Author Index