Computational Methods in Applied Sciences

Sergey Repin
Timo Tiihonen
Tero Tuovinen *Editors*

Numerical Methods for Differential Equations, Optimization, and Technological Problems

Dedicated to Professor P. Neittaanmäki on His 60th Birthday

ECCOMAS
European Community on Computational Methods in Applied Sciences

Springer

Numerical Methods for Differential Equations, Optimization, and Technological Problems

Computational Methods in Applied Sciences

Volume 27

Series Editor

E. Oñate
International Center for Numerical Methods in Engineering (CIMNE)
Technical University of Catalonia (UPC)
Edificio C-1, Campus Norte UPC
Gran Capitán, s/n
08034 Barcelona, Spain
onate@cimne.upc.edu
www.cimne.com

For further volumes:
www.springer.com/series/6899

Sergey Repin • Timo Tiihonen • Tero Tuovinen
Editors

Numerical Methods for Differential Equations, Optimization, and Technological Problems

Dedicated to Professor P. Neittaanmäki on His 60th Birthday

Springer

Editors
Sergey Repin
Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland

Tero Tuovinen
Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland

Timo Tiihonen
Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland

ISSN 1871-3033 Computational Methods in Applied Sciences
ISBN 978-94-007-5287-0 ISBN 978-94-007-5288-7 (eBook)
DOI 10.1007/978-94-007-5288-7
Springer Dordrecht Heidelberg New York London

Library of Congress Control Number: 2012948397

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Dedicated to Professor Pekka Neittaanmäki on the occasion of his 60th birthday

Foreword

This book exposes the results in numerical analysis and optimization presented at the ECCOMAS thematic conference "*Computational Analysis and Optimization*" (CAO 2011) in Jyväskylä, Finland, on June 9–11, 2011, dedicated to the 60th jubilee of Professor Pekka Neittaanmäki. It consists of five parts, which are closely related to his scientific activities and interests.

Part I of the book contains new results associated with numerical analysis of nonlinear models in continuum mechanics. It begins with a paper due to R.H.W. Hoppe and C. Linsenmann, in which the authors study the motion of red blood cells (RBCs) subject to an external incompressible flow in a microchannel and investigate two versions (semiexplicit and fully implicit) of the Finite Element Immersed Boundary Method (FE-IB). The paper by E. Laitinen and A. Lapin is focused on iteration methods for saddle-point problems. Such problems often arise in analysis of nonlinear models associated with variational inequalities.

A survey devoted to analytic-numerical methods for hidden attractors localization and their application to nonlinear dynamic systems is presented in a paper by G. Leonov and V. Kuznetsov. In a paper by S. Turek et al., the authors consider new numerical schemes created to simulate a 3D multiphase flow of immiscible fluids. The article of H. Wang and J. Periaux is concerned with a fast meshless method for problems in computational fluid dynamics.

Part II collects papers associated with the topic "a posteriori error estimates and reliable computer simulation methods". The first paper of R. Rannacher and J. Vihharev discusses an important question of balancing discretization and integration errors in numerical solutions computed by finite element methods. Analysis of this question is based on dual-weighted residual a posteriori estimates. In a paper due to S. Repin and T. Rossi, the authors use another class of a posteriori estimates (estimates of the functional type) in order to reformulate the classical inverse problem in the form of an unconstrained minimization problem. A posteriori estimates of the same type are derived in the next paper, where O. Mali considers higher-order boundary value problems in the theory of curvilinear beams. One more method of deriving a posteriori estimates is used in the article by S. Matsulevich et al. It is based on the theory of contractive mappings and is applied to the Picard-Lindelöf

method. Finally, the paper of K. Segeth contains an overview of a posteriori estimates for the biharmonic problem.

Part III contains publications related to optimization methods. It starts with a paper by M.M. Mäkelä et al., where the reader will find a new subgradient-type method of minimizing nonconvex functionals. In the next paper, written by P. Philip and D. Tiba, the authors discuss shape optimization problems governed by linear or nonlinear elliptic partial differential equations via a fixed domain approach. Methods of structural optimization in the application to biological models are discussed in the paper of M. Nowak.

Part IV is concerned with analysis of "noisy" and uncertain data. The paper of J.-A. Désidéri et al. investigates approximation properties of the Hermitian interpolation of a function whose nodal values are known exactly and the derivatives only approximately. The results can be used in analysis of the so-called "meta-models" arising in the theory of design of an experiment. The paper by A. Zerbinati et al. addresses the same topic. In a paper by A. Averbuch et al., the authors present a robust algorithm starting from 1D or 2D discrete noised data to approximately invert the heat equation. The paper of G. Wolf et al. is devoted to clustering and distance analysis of data sets.

Part V is focused on problems in paper machine industry and information technology. The paper by N. Banichuk and S. Ivanova studies the effects caused by uncertainties in contact mechanics and shape optimization problems. The paper by J. Jeronen addresses, in principle, the same topic (analysis of uncertainties in a real-life mechanical problem), but from a different point of view. Two other papers of the same research group (N. Banichuk et al. and T. Saksa et al.) are devoted to safety analysis and optimization of a moving elastic web travelling between two rollers at a constant axial velocity. The paper of A. Averbuch et al. is devoted to network models arising in information systems. Finally, the paper of J. Hartikainen et al. considers computer simulation methods for highly nonlinear models in solid mechanics.

In all the parts, we first present papers of invited speakers and then contributed papers joined by a common topic. We would like to thank all the authors for their contributions. All the papers included in the volume have been reviewed (normally by two independent reviewers) and many of them have been modified in accordance with the comments received. We would like to thank all the reviewers for their work, which made it possible to essentially improve some publications.

The editors are grateful to all the authors for their contributions and to the Federation of Finnish Learned Societies for financial support. We would like to thank Marja-Leena Rantalainen for her careful work on preparing the electronic version of the book and express sincere thanks to our counterpart in Springer.

Jyväskylä, Finland

Sergey Repin
Timo Tiihonen
Tero Tuovinen

Preface

At the occasion of the 60th birthday of Professor Pekka Neittaanmäki it is time to take a look at his versatile activities from the decades passed so far.

Young Athlete

Pekka was born in 1951 in Saarijärvi, Central Finland—a rural community associated with perseverance and a spirit of hope even under the hardest of conditions in Finnish poetry. Saarijärvi is also known for its active society in athletics which made sports a natural target for Pekka's energy in the teen years. His ongoing running and skiing activities stem from those times. His more extreme hobbies from those times are perhaps less known. Pole vaulting and ski jumping are certainly not that frequent in academic CVs. However, they are good indicators of Pekka's daring attitude.

Emergence of a Researcher

In early 70's Pekka entered the University of Jyväskylä in the early 70's (with intention to graduate fast as math's teacher). Despite of the desire to graduate fast, Pekka did not ignore the social aspects of life as a student, which in the seventies implied also political activity, and gathered a strong personal network of future decision makers that has been an essential asset ever since. Fast graduation caught the attention of Professor I.S. Louhivaara who recruited Pekka for doctoral studies. Louhivaara, who had been trained as the right hand of Rolf Nevanlinna, had an extensive and active international network and a broad understanding of the active fields in mathematics. Thus, in his thesis work on boundary value problems to plate equation, mentored by K. Witsch from Bonn, Pekka got an extensive background in mathematical analysis of PDEs.

After his PhD Pekka went to Bonn as a post doc and got acquainted with both a generation of young German researchers and, more importantly, the finite element

method. When Pekka returned from Bonn, he brought a pile of punch cards containing a FEM code and several manuscripts and ideas to be written jointly with R. Picard and J. Saranen on FEM for different wave and eigenvalue problems.

Building His Own Group

In the beginning of the 80's Pekka started to be on his own. The FEM papers after the PhD qualified him to the level of a docent and it was time to start teaching and recruiting students. The first seminar around FEM was organized in Jyväskylä in fall 1982 and it attracted the attention of several promising students. Pekka exploited well the position of Finland as a gateway between East and West and quickly built a collaboration network with Czech and Romanian researchers. This helped him to both deepen and broaden the research activities. By a fortunate mishap Pekka learned about the superconvergence phenomenon related to averaging and managed to analyze it with M. Krizek with whom he continued the FEM track to review articles and eventually to monographs.

Simultaneously Pekka found other long-lasting interests: shape optimisation with J. Haslinger and optimal control with D. Tiba, both of these, in particular, in the context of variational inequalities. These openings led not only to numerous papers and several monographs in the years to come but also to a fair number of PhD students and theses, and to the first major industrial collaboration in the field of steel casting. The collaboration with industry was facilitated by Pekka's short excursion to Lappeenranta University of Technology as an associate professor and by a membership in the Research Council of Technology.

To facilitate his return back to Jyväskylä, Pekka played a major role in the University's initiative of "Applied Science" and managed to raise significant donations from the region to open new positions in computational and applied sciences, including the professorship he has been holding from 1988.

Shaping the University

In the beginning of the 90's the activities were booming. Being established as full professor, knowing the research councils from inside, having a successful industrial case in his portfolio and several young post docs, Pekka could expand the group with new students and attract new industrial partners and international collaborators. Simultaneously the situation in Eastern Europe had changed. Pekka was fast to make good contacts in the leading schools in St. Petersburg and Moscow while summarizing the fruits of collaboration with colleagues from Eastern Europe and establishing links to the French school of applied mathematics and the European free boundary community. This all has made Jyväskylä a collaboration hub for scientists from many countries. Simultaneously Pekka was busy internationalizing the group, by co-initiating the Jyväskylä International Summer Semester, starting the

organizing of international conferences and by recruiting first foreign PhD students to the group. This alone would have been more than enough to fill the days of a professor. But other things emerged in parallel.

Pekka was called to serve in a government-level committee of research issues and, after a few months as Dean of the Faculty of Natural Sciences, as the first vice-rector of the University. The period as vice-rector manifested Pekka's ability to recognize qualitatively new opportunities as they emerge. Finland, by joining the European Union in the beginning of 1995, became eligible to European regional funding—Jyväskylä region, in particular, because of a major crisis in the regional economy. Thanks to Pekka's initiatives (and personal network of strategic people), Jyväskylä was able to harness the EU funding to a real structural change in the local economy, largely through expansion and modernization of university education by targeted master programs.

The master programs and other actions expanded the IT-related activities so that they fitted neither in the facilities nor the organization of the University. This lead to the creation of the Faculty of Information Technology, splitting up the Department of Mathematics, and forming the Department of Mathematical Information Technology. Simultaneously a new building was needed and Pekka was a key person in the conceptualization of a site that linked several academic disciplines and related enterprises under a common roof. To implement the idea timely, new types of financial instruments had to be piloted at the same time. All this helped the University to move qualitatively forward to the new millennium.

Collaboration Across the Disciplines

In 2000 Pekka turned towards new challenges. Giving up the vice-rector duties, he started to promote collaboration between computational sciences and human oriented sciences actively. He saw the need of a platform for multidisciplinary research and collaboration and helped to establish Agora Center as such a unit. Serving several years as the head of Agora Center, Pekka has helped several multidisciplinary groups to find collaborators and new funding opportunities. Once again Pekka has been alert and able to react to the changes in the society and our region by launching an impressive series of actions to counterbalance the effects of the global financial crisis of 2008 by targeted educational and research programs.

To parallel his activities in multidisciplinary research, Pekka has expanded his personal research interests to game theory, data mining, and the like. When doing this he has not forgotten his roots. His first love, plate equation and wave phenomena, has led him to active collaboration and contributions in the field of nanotechnology. Error estimates for finite elements are still a relevant topic, now in the context of reliable a posteriori estimates.

Pekka has continued to be active in fatherly supervision of his PhD students, now passing over 60 in cumulative count.

60 and Beyond

What can be said about the future? Predicting the behavior of multidimensional dynamic systems is never easy. In Pekka's case it is virtually impossible. Having grown up with exterior problems, Pekka sees his domain unbounded by nature. He knows that irregularities from incompatibilities tend to smooth out asymptotically and that obstacles and barriers can be overcome, perhaps with paying a small penalty. He knows that even non-smooth systems can be controlled and optimized, that optimal solutions may be structurally different from the current design and that approximate solutions will do for practical cases. So, honestly, we cannot predict what exactly Pekka will be doing in the future.

However, 60 years of observation is enough to infer what Pekka will be working for—for the benefit of his friends and collaborators, for the University of Jyväskylä, for the region of Central Finland, for Finland, and beyond.

Jyväskylä, Finland Timo Tiihonen

Contents

Contributors

Tuomas Airaksinen Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Timo Aittokoski Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Amir Averbuch School of Computer Science, Tel Aviv University, Tel Aviv, Israel; Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Adil Bagirov Centre for Informatics and Applied Optimization, School of Science, Information Technology and Engineering, University of Ballarat, Ballarat, VIC, Australia

Nikolay Banichuk Ishlinsky Institute for Problems in Mechanics, Russian Academy of Sciences (RAS), Moscow, Russia

Manuel Bompard ONERA/DSNA, Châtillon cedex, France

Hong-Quan Chen Department of Aerodynamics, Nanjing University of Aeronautics and Astronautics, Nanjing, P.R. China

Jean-Antoine Désidéri INRIA, Centre de Sophia Antipolis – Méditerranée, Sophia Antipolis cedex, France; INRIA, Sophia Antipolis, France

Régis Duvigneau INRIA, Sophia Antipolis, France

Juha Hartikainen Aalto University, Aalto, Finland

Shachar Harussi School of Computer Science, Tel Aviv University, Tel Aviv, Israel; Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Ronald H.W. Hoppe Institute of Mathematics, Universität Augsburg, Augsburg, Germany; Department of Mathematics, University of Houston, Houston, TX, USA

Shuren Hysing Department of Mathematics, Shanghai Jiaotong University, Shanghai, China

Svetlana Ivanova Ishlinsky Institute for Problems in Mechanics, Russian Academy of Sciences, Moscow, Russia

Juha Jeronen Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Napsu Karmitsa Department of Mathematics, University of Turku, Turku, Finland

Kari Kolari VTT, VTT, Finland

Reijo Kouhia Tampere University of Technology, Tampere, Finland

Matti Kurki Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland; School of Technology, JAMK University of Applied Sciences, Jyväskylä, Finland

Dmitri Kuzmin Lehrstuhl für Angewandte Mathematik III, Universität Erlangen-Nürnberg, Erlangen, Germany

Nikolay V. Kuznetsov St. Petersburg State University, St. Petersburg, Russia; Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Erkki Laitinen University of Oulu, Oulu, Finland

Alexander Lapin Kazan Federal University, Kazan, Russia

Gennady A. Leonov St. Petersburg State University, St. Petersburg, Russia

Christopher Linsenmann Institute of Mathematics, Universität Augsburg, Augsburg, Germany

Olli Mali Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Svetlana Matculevich Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Marko M. Mäkelä Department of Mathematics, University of Turku, Turku, Finland

Otto Mierka Institut für Angewandte Mathematik, TU Dortmund, Dortmund, Germany

Pekka Neittaanmäki Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Michal Nowak Department of Machine Design Method, Poznan University of Technology, Poznan, Poland

Jacques Periaux Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland; International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain

Jacques Peter ONERA/DSNA, Châtillon cedex, France

Peter Philip Department of Mathematics, Ludwig-Maximilians University (LMU) Munich, Munich, Germany

Rolf Rannacher Institute of Applied Mathematics, University of Heidelberg, Heidelberg, Germany

Sergey Repin Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland; V.A. Steklov Institute of Mathematics in St. Petersburg, St. Petersburg, Russia

Yehuda Roditi Academic College of Tel-Aviv-Yaffo, Tel Aviv, Israel

Tuomo Rossi Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Tytti Saksa Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Karel Segeth Institute of Mathematics, Academy of Sciences, Prague, Czech Republic

Yaniv Shmueli School of Computer Science, Tel Aviv University, Tel Aviv, Israel

Dan Tiba Institute of Mathematics, Romanian Academy, Bucharest, Romania; Academy of Romanian Scientists, Bucharest, Romania

Maria Tirronen Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Tero Tuovinen Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Stefan Turek Institut für Angewandte Mathematik, TU Dortmund, Dortmund, Germany

Jevgeni Vihharev Institute of Applied Mathematics, University of Heidelberg, Heidelberg, Germany

Hong Wang Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Guy Wolf School of Computer Science, Tel Aviv University, Tel Aviv, Israel; Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Nezer Jacob Zaidenberg Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Adrien Zerbinati INRIA, Sophia Antipolis, France

Valery Zheludev School of Computer Science, Tel Aviv University, Tel Aviv, Israel; Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Part I
Numerical Methods for Nonlinear Problems

Chapter 1
The Finite Element Immersed Boundary Method for the Numerical Simulation of the Motion of Red Blood Cells in Microfluidic Flows

Ronald H.W. Hoppe and Christopher Linsenmann

Abstract We study the mathematical modeling and numerical simulation of the motion of red blood cells (RBCs) subject to an external incompressible flow in a microchannel. RBCs are viscoelastic bodies consisting of a deformable elastic membrane enclosing an incompressible fluid. We study two versions of the Finite Element Immersed Boundary Method (FE-IB), a semi-explicit scheme that requires a CFL-type stability condition and a fully implicit scheme that is unconditionally stable and numerically realized by a predictor-corrector continuation strategy featuring an adaptive choice of the time step sizes. The performance of the two schemes is illustrated by numerical simulations for various scenarios including the tank treading motion in microchannels and the motion through thin capillaries.

Keywords Microfluidic flows · Red blood cells · Finite element immersed boundary method · Semi-explicit scheme · Fully implicit scheme

1.1 Introduction

Red blood cells are viscoelastic bodies which, roughly speaking, consist of a membrane enclosing a liquid [2, 6–9, 26, 30, 31]. When exposed to an external flow, the motion in the fluid represents a fluid-structure interaction problem that can be appropriately modeled by the finite element immersed boundary method (FE-IB). As opposed to the classical immersed boundary method (IB) [24, 25], which is based on a finite difference approach, the FE-IB relies on the variational formulation of the problem [3, 4]. We consider a semi-discretization in space by using Taylor–Hood

R.H.W. Hoppe (✉) · C. Linsenmann
Institute of Mathematics, Universität Augsburg, 86159 Augsburg, Germany
e-mail: hoppe@math.uni-augsburg.de

C. Linsenmann
e-mail: christopher.linsenmann@math.uni-augsburg.de

R.H.W. Hoppe
Department of Mathematics, University of Houston, Houston, TX 77204-3008, USA
e-mail: rohop@math.uh.edu

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_1,

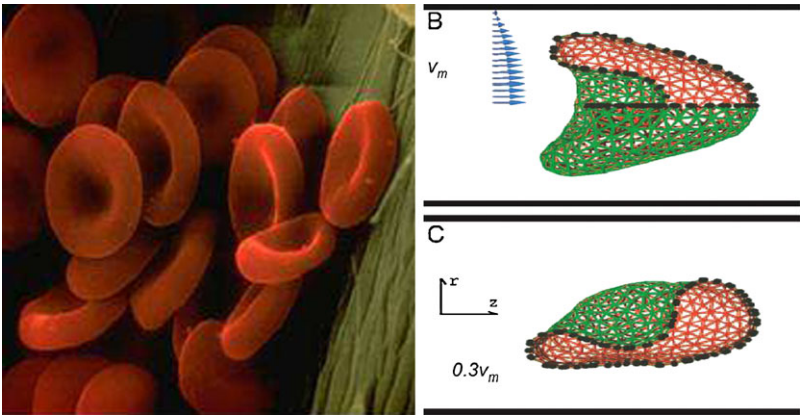


Fig. 1.1 Equilibrium configuration of RBCs (*left*) and deformed shapes in Poiseuille flow (*right*)

P2–P1 elements for the incompressible Navier–Stokes equations and periodic cubic splines for the immersed boundaries. For discretization in time we distinguish between the semi-explicit Backward Euler/Forward Euler FE-IB and the fully implicit Backward Euler/Backward Euler FE-IB where in both cases we use implicit time-stepping for the semi-discretized Navier–Stokes equations. On the other hand, the IB equations are discretized by the forward Euler method in the BE/FE FE-IB and by the backward Euler method in the fully implicit BE/BE FE-IB. The BE/FE FE-IB is subject to a CFL-type condition, whereas the BE/BE FE-IB is unconditionally stable. The latter one will be solved by a predictor-corrector continuation strategy featuring an adaptive choice of the continuation parameter. For both schemes we provide a documentation of numerical results including a comparison with experimental data.

1.2 The Finite Element Immersed Boundary Method

As is well-known, human blood is a suspension of viscoelastic cells, the red and white blood cells, in a viscous fluid, the plasma. The blood flow is not only controlled by the viscosity of the plasma, but additionally viscoelastic effects due to the deformability of the cells, aggregation of the cells, and hematocrit come into play and have been experimentally observed.

If not being subjected to an external flow, red blood cells (RBCs) are biconcavely shaped disks with a diameter of 7.5–8.0 μm and a thickness of about 2 μm (cf. Fig. 1.1 (left)). However, under the influence of an external flow, the cells deform and may attain parachute- or slipper-like shapes in Poiseuille flow depending on their position (centered or decentered) in the flow field [22] (cf. Fig. 1.1 (right)).

The FE-IB describes the fluid-structure interaction problem based on a coupled Eulerian/Lagrangian approach. The external fluid flow is modeled by the incompressible Navier–Stokes equations in an Eulerian coordinate system, whereas the

boundaries of the immersed bodies are described by means of a Lagrangian coordinate system. For its variational formulation, we will use standard notation from Lebesgue and Sobolev space theory [29].

In particular, we assume the computational domain $\Omega := (a,b) \times (c,d)$, $a < b$, $c < d$, to be a microchannel with boundary $\Gamma := \bar{\Gamma}_{\text{in}} \cup \bar{\Gamma}_{\text{out}} \cup \bar{\Gamma}_{\text{lat}}$, where $\Gamma_{\text{in}} := \{a\} \times (c,d)$, $\Gamma_{\text{out}} := \{b\} \times (c,d)$, and $\Gamma_{\text{lat}} := (a,b) \times \{c\} \cup (a,b) \times \{d\}$. We further assume a stationary velocity g at the inflow boundary Γ_{in} and the outflow boundary Γ_{out} as well as zero velocity at the lateral boundary Γ_{lat}. The impact of the immersed bodies on the external force is described by a force density $\mathbf{F}$ which will be specified later. Denoting by ρ, μ the density and the viscosity of the external fluid, by $\mathbf{u}$, p the velocity and the pressure, by $\mathbf{u}^{(0)}$ an initial velocity, and by $\mathbf{n}_{\Sigma_{\text{in}}}$, $\mathbf{n}_{\Sigma_{\text{out}}}$, $\mathbf{t}_{\Sigma_{\text{in}}}$, $\mathbf{t}_{\Sigma_{\text{out}}}$ the exterior unit normal vectors and unit tangential vectors on Σ_{in}, Σ_{out}, the incompressible Navier–Stokes equations read

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{v}\right) - \mu\Delta\mathbf{u} + \nabla p = \mathbf{F} \qquad \text{in } Q := \Omega \times (0,T], \tag{1.1a}$$

$$\nabla\cdot\mathbf{u} = 0 \qquad \text{in } Q, \tag{1.1b}$$

$$\mathbf{t}_{\Sigma_{\text{in}}}\cdot\mathbf{u} = 0, \quad \mathbf{n}_{\Sigma_{\text{in}}}\cdot\mathbf{u} = g \qquad \text{on } \Sigma_{\text{in}} := \Gamma_{\text{in}} \times (0,T], \tag{1.1c}$$

$$\mathbf{t}_{\Sigma_{\text{out}}}\cdot\mathbf{u} = 0, \quad \mathbf{n}_{\Sigma_{\text{out}}}\cdot\mathbf{u} = g \qquad \text{on } \Sigma_{\text{out}} := \Gamma_{\text{out}} \times (0,T], \tag{1.1d}$$

$$\mathbf{u} = 0 \qquad \text{on } \Sigma_{\text{lat}} := \Gamma_{\text{lat}} \times (0,T], \tag{1.1e}$$

$$\mathbf{u}(\cdot,0) = \mathbf{u}^{(0)} \qquad \text{in } \Omega. \tag{1.1f}$$

The boundary of an immersed body is supposed to be a smooth closed and non-intersecting, massless curve of length L driven by the velocity $\mathbf{u}$. Denoting by $\mathbf{X}$ the position vector of the boundary, by $\mathbf{X}^{(0)}$ the initial configuration, and assuming periodic boundary conditions, the equations of motion of the immersed body read

$$\frac{\partial \mathbf{X}}{\partial t} = \mathbf{u}\big(\mathbf{X}(\cdot,t),t\big), \tag{1.2a}$$

$$\mathbf{X}(q,0) = \mathbf{X}^{(0)}(q), \qquad q \in [0,L], \tag{1.2b}$$

$$\partial^k\mathbf{X}/\partial q^k(0,t) = \partial^k\mathbf{X}/\partial q^k(L,t), \quad k = 0,1,2. \tag{1.2c}$$

The force density $\mathbf{F} \in \mathbf{L}^2((0,T),\mathbf{H}^{-1}(\Omega))$ in (1.1a) is given by

$$\big\langle \mathbf{F}(t),\mathbf{v}\big\rangle = \int_0^L \mathbf{f}(q,t)\cdot\mathbf{v}\big(\mathbf{X}(q,t)\big)\,dq \quad \text{for almost all } t \in (0,T). \tag{1.3}$$

Here, $\langle\cdot,\cdot\rangle$ stands for the dual pairing between $\mathbf{H}^{-1}(\Omega)$ and $\mathbf{H}_0^1(\Omega)$. The local force density $\mathbf{f}$ is defined according to $\mathbf{f}(q,t) = -E'(\mathbf{X}(q,t))$ by means of the Gâteaux derivative E' of the total energy $E(t) := E^e(t) + E^b(t)$, where

$$E^e(t) := \int_0^L \frac{\kappa_e}{2}\left(\left|\frac{\partial \mathbf{X}}{\partial q}(q,t)\right| - 1\right)^2 dq, \tag{1.4a}$$

$$E^b(t) := \int_0^L \frac{\kappa_b}{2} \left| \frac{\partial^2 \mathbf{X}}{\partial q^2}(q,t) \right|^2 dq. \tag{1.4b}$$

We introduce the function spaces

$$\begin{aligned}
\mathbf{V}(0,T) &:= \mathbf{H}^1\big((0,T), \mathbf{H}^{-1}(\Omega)\big) \cap \mathbf{L}^2\big((0,T), \mathbf{H}^1(\Omega)\big),\\
\mathbf{W}(0,T) &:= \big\{\mathbf{w} \in \mathbf{V}(0,T) \mid \mathbf{n}_{\Sigma_{\text{in}}} \cdot \mathbf{w}|_{\Sigma_{\text{in}}} = \mathbf{n}_{\Sigma_{\text{out}}} \cdot \mathbf{w}|_{\Sigma_{\text{out}}} = g,\\
&\qquad \mathbf{t}_{\Sigma_{\text{in}}} \cdot \mathbf{w}|_{\Sigma_{\text{in}}} = \mathbf{t}_{\Sigma_{\text{out}}} \cdot \mathbf{w}|_{\Sigma_{\text{out}}} = 0,\ \mathbf{w}|_{\Sigma_{\text{lat}}} = 0\big\},\\
\mathbf{W}_0(0,T) &:= \big\{\mathbf{w} \in \mathbf{V}(0,T) \mid \mathbf{w}_{\Gamma \times (0,T]} = \mathbf{0}\big\},\\
Q(0,T) &:= L^2\big((0,T), L_0^2(\Omega)\big).
\end{aligned}$$

The variational formulation of the FE-IB amounts to the computation of

$$(\mathbf{u}, p, \mathbf{X}) \in \mathbf{W}(0,T) \times Q(0,T) \times \mathbf{H}^1\big((0,T), \mathbf{L}^2\big([0,L]\big)\big) \cap \mathbf{L}^2\big((0,T), \mathbf{H}^3_{per}\big([0,L]\big)\big)$$

such that for all $(\mathbf{w}, q) \in \mathbf{W}_0(0,T) \times Q(0,T)$ there holds

$$\left\langle \rho \frac{\partial \mathbf{u}}{\partial t}, \mathbf{w} \right\rangle + a(\mathbf{u}, \mathbf{w}) - b(p, \mathbf{w}) = \big\langle \mathbf{F}(t), \mathbf{w} \big\rangle, \tag{1.5a}$$

$$b(q, \mathbf{u}) = 0, \tag{1.5b}$$

$$\mathbf{u}(\cdot, 0) = \mathbf{u}^{(0)}, \tag{1.5c}$$

and $\mathbf{X}$ satisfies (1.2a)–(1.2c). Here, the forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are given by

$$a(\mathbf{u}, \mathbf{v}) := \big(\rho(\mathbf{u} \cdot \nabla)\mathbf{u}, \mathbf{v}\big)_{0,\Omega} + (\mu \nabla \mathbf{u}, \nabla \mathbf{v})_{0,\Omega}, \tag{1.6a}$$

$$b(p, \mathbf{v}) := (p, \nabla \cdot \mathbf{v})_{0,\Omega}. \tag{1.6b}$$

The following result provides an energy estimate for the FE-IB.

Theorem 1 *Suppose that the data of the FE-IB satisfy*

$$\mathbf{F} \in \mathbf{L}^2\big((0,T); \mathbf{H}^{-1}(\Omega)\big), \qquad \mathbf{u}^{(0)} \in \mathbf{L}^2(\Omega), \tag{1.7a}$$

$$g \in H_{00}^{5/2+s}(\Gamma_{\text{in}} \cup \Gamma_{\text{out}}), \qquad s \in (0, 1/2), \tag{1.7b}$$

and

$$\|g\|_{H_{00}^{5/2+s}(\Gamma_{\text{in}})} \le \big(4\rho C(\Omega)\big)^{-1} \mu, \quad s \in (0, 1/2), \tag{1.8}$$

for some constant $C(\Omega) > 0$, depending on the domain Ω. Then, if the triple $(\mathbf{u}, p, \mathbf{X})$ solves (1.5a)–(1.5c) and (1.2a)–(1.2c), there exist positive constants C_i, $1 \le i \le 2$, only depending on the data of the problem, and $C(g)$, only depending

on g, such that

$$\frac{\rho}{4}\left\|\mathbf{u}(\cdot,t)\right\|_{0,\Omega}^2+\frac{\mu}{4}\int_0^t\left\|\nabla\mathbf{u}(\cdot,\tau)\right\|_{0,\Omega}^2 d\tau+E(t)$$
$$\leq C_1\left(1+C_2\,t\;+\left\|\mathbf{u}^{(0)}\right\|_{0,\Omega}^2+E(0)+C(g)\int_0^t E(\tau)\,d\tau\right).$$

Proof We refer to [14]. □

1.3 Space/Time Discretization

For the spatial discretization of the incompressible Navier–Stokes equations we use P2–P1 Taylor–Hood elements [5] with respect to a simplicial triangulation $\mathscr{T}_h(\Omega)$ of the computational domain Ω. For $T\in\mathscr{T}_h(\Omega)$, we refer to h_T as the diameter of T and set $h:=\max_{T\in\mathscr{T}_h(\Omega)}h_T$. We further denote by $P_k(T)$, $k\in\mathbb{N}$, the linear space of polynomials of degree $\leq k$ on T, and we choose the function spaces

$$\mathbf{V}_h:=\left\{\mathbf{v}_h\in C(\bar{\Omega})^2\mid v_h|_T\in P_2(T)^2\right\},\qquad Q_h:=\left\{q_h\in C(\bar{\Omega})\mid q_h|_T\in P_1(T)\right\},$$

and

$$\begin{aligned}\mathbf{W}_h(0,T):=\big\{&\mathbf{w}_h\in C\big([0,T],C(\bar{\Omega})^2\big)\mid\mathbf{w}_h(\cdot,t)\in\mathbf{V}_h,\\&\mathbf{n}_{\Gamma_{\text{in}}}\cdot\mathbf{w}_h=\mathbf{n}_{\Gamma_{\text{out}}}\cdot\mathbf{w}_h=g_h,\\&\mathbf{t}_{\Gamma_{\text{in}}}\cdot\mathbf{w}_h=\mathbf{t}_{\Gamma_{\text{out}}}\cdot\mathbf{w}_h=0,\;\mathbf{n}_{\Gamma_{\text{lat}}}\cdot\mathbf{w}_h=0\big\},\\\mathbf{W}_{h,0}(0,T):=\big\{&\mathbf{w}_h\in\mathbf{W}_h(0,T)\mid\mathbf{w}_h|_{\Gamma\times(0,T)}=0\big\},\\Q_h(0,T):=\big\{&q_h\in C\big([0,T],C(\bar{\Omega})\cap L_0^2(\Omega)\big)\mid q_h(\cdot,t)\in Q_h\big\}.\end{aligned}$$

Given some approximation $g_h\in C([0,T],C^2(\Gamma_{\text{in}}))$ of the inflow velocity g and $\mathbf{u}_h^{(0)}\in\mathbf{V}_h$ of the initial velocity $\mathbf{u}^{(0)}$, we compute $(\mathbf{u}_h,p_h)\in\mathbf{W}_h(0,T)\times Q_h(0,T)$ such that for all $(\mathbf{w}_h,q_h)\in\mathbf{W}_{h,0}(0,T)\times Q_h(0,T)$ and $t\in[0,T]$ there holds

$$\left(\rho\frac{\partial\mathbf{u}_h}{\partial t},\mathbf{w}_h\right)_{0,\Omega}+a(\mathbf{u}_h,\mathbf{w}_h)-b(p_h,\mathbf{w}_h)=\left\langle\mathbf{F}_h(t),\mathbf{w}_h\right\rangle_h,\tag{1.9a}$$

$$b(q_h,\mathbf{u}_h)=0,\tag{1.9b}$$

$$\mathbf{u}_h(0,t)=\mathbf{u}_h^{(0)},\tag{1.9c}$$

where $\langle\mathbf{F}_h(t),\mathbf{w}_h\rangle_h$ will be defined by (1.11) below.

We discretize the immersed boundary by periodic cubic splines with respect to a partition

$$\mathscr{T}_{[0,L]}:=\{0=:q_0<q_1<\cdots<q_M:=L\},\quad M\in\mathbb{N},$$

of the interval $[0, L]$ into subintervals $I_i := [q_{i-1}, q_i]$ of length $\Delta q_i := q_i - q_{i-1}$, and we set

$$\mathbf{S}_h := \big\{\mathbf{Y}_h \in C^2\big([0, L], \Omega\big) \mid \mathbf{Y}_h|_{I_i} \in P_3(I_i)^2,\ \mathbf{Y}_h^{(k)}(q_0) = \mathbf{Y}_h^{(k)}(q_M),\ 0 \le k \le 2\big\}.$$

Given some approximation $\mathbf{X}_h^{(0)} \in \mathbf{S}_h$ of $\mathbf{X}^{(0)}$, we look for $\mathbf{X}_h \in C^1([0, T], \mathbf{S}_h)$ such that

$$\frac{\partial \mathbf{X}_h}{\partial t} = \mathbf{u}_h\big(\mathbf{X}_h(\cdot, t), t\big), \quad 0 < t \le T, \tag{1.10a}$$

$$\mathbf{X}_h(\cdot, 0) = \mathbf{X}_h^{(0)}. \tag{1.10b}$$

Finally, the right-hand side in (1.9a) reads as follows:

$$\begin{aligned}\big\langle \mathbf{F}_h(t), \mathbf{w}_h\big\rangle_h = &-\kappa_e \int_0^L \frac{\partial \mathbf{X}_h}{\partial q} \cdot \nabla \mathbf{w}_h\big(\mathbf{X}_h(q, t)\big)\, \frac{\partial \mathbf{X}_h}{\partial q}\, dq \\ &+ \kappa_b \sum_{i=1}^M \frac{\partial^3 \mathbf{X}_h}{\partial q^3}\bigg|_{I_i} \cdot \int_{q_{i-1}}^{q_i} \nabla \mathbf{w}_h\big(\mathbf{X}_h(q, t)\big) \frac{\partial \mathbf{X}_h}{\partial q}\, dq. \end{aligned} \tag{1.11}$$

For discretization in time, we consider a partition $\mathcal{T}_{[0,T]}$ of the time interval $[0, T]$

$$\mathcal{T}_{[0,T]} := \{0 =: t_0 < t_1 < \cdots < t_N := T\}, \quad N \in \mathbb{N},$$

into subintervals $[t_n, t_{n+1}]$, $0 \le n \le N-1$, of length $\tau_n := t_{n+1} - t_n$. We denote by $\mathbf{u}_h^{(n)}$ approximations of $\mathbf{u}_h$ at times t_n and define

$$D_{\Delta t}^+ \mathbf{u}_h^{(n)} := \frac{\mathbf{u}_h^{(n+1)} - \mathbf{u}_h^{(n)}}{\Delta \tau_n}.$$

Semi-explicit Scheme The Backward Euler/Forward Euler FE-IB (BE/FE FE-IB) is a semi-explicit scheme where we discretize the semi-discrete Navier–Stokes equations (1.9a)–(1.9c) by the implicit Euler scheme and the immersed boundary equations (1.10a), (1.10b) by the explicit Euler scheme. In particular, given $\mathbf{u}_h^{(0)}$, for $n \ge 0$ we first compute $(\mathbf{u}_h^{(n+1)}, p_h^{(n+1)}) \in \mathbf{V}_h \times Q_h$ such that

$$\big(\rho\ D_{\Delta t}^+ \mathbf{u}_h^{(n)}, \mathbf{w}_h\big)_{0,\Omega} + a\big(\mathbf{u}_h^{(n+1)}, \mathbf{w}_h\big) - b\big(p_h^{(n+1)}, \mathbf{w}_h\big) = \big\langle \mathbf{F}_h^{(n)}, \mathbf{w}_h\big\rangle_h, \tag{1.12a}$$

$$b\big(q_h, \mathbf{u}_h^{(n+1)}\big) = 0, \tag{1.12b}$$

where

$$\langle \mathbf{F}_h^{(n)}, \mathbf{w}_h \rangle_h := -\kappa_e \int_0^L \frac{\partial \mathbf{X}_h^{(n)}}{\partial q} \cdot \frac{\partial}{\partial q} \mathbf{w}_h\big(\mathbf{X}_h^{(n)}\big)\, dq + \kappa_b \sum_{i=1}^{M} \left. \frac{\partial^3 \mathbf{X}_h^{(n)}}{\partial q^3} \right|_{I_i} \cdot \int_{q_{i-1}}^{q_i} \frac{\partial}{\partial q} \mathbf{w}_h\big(\mathbf{X}_h^{(n)}\big)\, dq$$

and then compute $\mathbf{X}_h^{(n+1)} \in \mathbf{S}_h$ according to

$$\int_0^L \mathbf{X}_h^{(n+1)} \cdot \mathbf{Y}_h\, dq = \tau_n \int_0^L \mathbf{u}_h^{(n)}\big(\mathbf{X}_h^{(n)}\big) \cdot \mathbf{Y}_h\, dq + \int_0^L \mathbf{X}_h^{(n)} \cdot \mathbf{Y}_h\, dq, \quad \mathbf{Y}_h \in \mathbf{S}_h. \tag{1.13}$$

In case of an equidistant partition $\mathscr{T}_{[0,T]}$ with $\tau_n = \Delta t := T/N$, the following stability estimate holds true (cf. [14]).

Theorem 2 *In addition to the assumptions of Theorem 1 suppose that the CFL-type condition*

$$\frac{\Delta\tau}{h} \le \frac{\mu}{4C_B(\kappa_e L_1 + \kappa_b L_2)}, \tag{1.14}$$

is satisfied, where $C_B > 0$ is a computable constant, depending on the domain Ω, and the constants L_i, $1 \le i \le 2$, are given by

$$\begin{aligned} L_1 &:= \max_{0\le n\le N} \max_{q\in[0,L]} \big|\partial \mathbf{X}_h^{(n)}/\partial q\big|, \\ L_2 &:= \max_{0\le n\le N} \max_{1\le i\le M} \max_{q\in I_i} \big|\partial^3 \mathbf{X}_h^{(n)}/\partial q^3|_{I_i}\big|. \end{aligned} \tag{1.15}$$

Then, if $(\mathbf{u}_h, p_h, \mathbf{X}_h)$ solves the BE/FE FE-IB (1.12a), (1.12b), (1.13), there exist positive constants C_i, $4 \le i \le 6$, depending only on the data of the problem, such that

$$\begin{aligned} &\frac{\rho}{4} \big\|\mathbf{u}_h^{(n)}\big\|_{0,\Omega}^2 + \frac{\mu}{4} \sum_{m=1}^{n} \big\|\nabla \mathbf{u}_h^{(m)}\big\|_{0,\Omega}^2 \Delta\tau + E_h^{(n)} \\ &\quad \le C_4 \left(1 + C_5\, t_n + \big\|\mathbf{u}_h^{(0)}\big\|_{0,\Omega}^2 + E_h^{(0)} + C_6 \sum_{m=1}^{n-1} E_h^{(m)} \Delta\tau \right). \end{aligned} \tag{1.16}$$

Remark 1 Semi-explicit schemes based on the classical IB have been applied to the simulation of the motion of RBCs in [11, 23, 33].

Fully Implicit Scheme The Backward Euler/Backward Euler FE-IB (BE/BE FE-IB) is a fully implicit scheme where we discretize both the semi-discrete Navier–Stokes equations (1.9a)–(1.9c) and the immersed boundary equations (1.10a),

(1.10b) by the backward Euler scheme. In particular, we simultaneously compute $(\mathbf{u}_h^{(n+1)}, p_h^{(n+1)}) \in \mathbf{V}_h \times Q_h$ such that for all $\mathbf{w}_h \in \mathbf{W}_{h,0}(0,T)$, $q_h \in Q_h$ it holds

$$\big(\rho\, D_{\tau_n}^{+}\mathbf{u}_h^{(n)}, \mathbf{w}_h\big)_{0,\Omega} + a\big(\mathbf{u}_h^{(n+1)}, \mathbf{w}_h\big) - b\big(p_h^{(n+1}, \mathbf{w}_h\big) = \big\langle \mathbf{F}_h^{(n+1)}, \mathbf{w}_h\big\rangle_h, \tag{1.17a}$$

$$b\big(q_h, \mathbf{u}_h^{(n+1)}\big) = 0, \tag{1.17b}$$

where

$$\begin{aligned}\big\langle \mathbf{F}_h^{(n+1)}, \mathbf{w}_h\big\rangle_h := &-\kappa_e \int_0^L \frac{\partial \mathbf{X}_h^{(n+1)}}{\partial q} \cdot D^1\mathbf{w}_h\big(\mathbf{X}_h^{(n+1)}\big)\frac{\partial \mathbf{X}_h^{(n+1)}}{\partial q}\, dq \\ &+ \kappa_b \int_0^L \frac{\partial^2 \mathbf{X}_h^{(n+1)}}{\partial q^2} \cdot \bigg(D^1\mathbf{w}_h\big(\mathbf{X}_h^{(n+1)}\big)\frac{\partial^2 \mathbf{X}_h^{(n+1)}}{\partial q^2} \\ &+ D^2\mathbf{w}_h\big(\mathbf{X}_h^{(n+1)}\big)\bigg(\frac{\partial \mathbf{X}_h^{(n+1)}}{\partial q}, \frac{\partial \mathbf{X}_h^{(n+1)}}{\partial q}\bigg)\bigg)\, dq,\end{aligned}$$

and compute $\mathbf{X}_h^{(n+1)} \in \mathbf{S}_h$ such that for all $\mathbf{Y}_h \in \mathbf{S}_h$ there holds

$$\int_0^L \mathbf{X}_h^{(n+1)} \cdot \mathbf{Y}_h\, dq - \tau_n \int_0^L \mathbf{u}_h^{(n+1)}\big(\mathbf{X}_h^{(n+1)}\big) \cdot \mathbf{Y}_h\, dq = \int_0^L \mathbf{X}_h^{(n)} \cdot \mathbf{Y}_h\, dq. \tag{1.18}$$

Remark 2 For the classical IB, semi-explicit, approximate implicit, and fully implicit schemes have been considered in [18–21, 28], whereas the unconditional stability of fully implicit FE-IB methods has been shown in [4].

1.4 Predictor–Corrector Continuation Strategy for the Numerical Solution of the Fully Implicit Scheme

The numerical realization of the fully implicit BE/BE FE-IB amounts to the solution of a nonlinear system of equations. The application of Newton's method turns out to be delicate, since the numerical stiffness of the BE/BE FE-IB significantly affects the convergence radius of Newton's method in a negative way. In compliance with this, for the fully implicit IB it was stated in [32] that Newton's method is computationally too expensive in practice. To overcome this difficulty, we will use a continuation method in a predictor–corrector manner. Thereby, the time increment is chosen adaptively in such a way that the convergence requirements of Newton's method for the next time step are met. Consequently, a successful application of Newton's method is guaranteed without an expensive search for proper initial guesses.

At each time-step, the BE/BE FE-IB (1.17a), (1.17b), (1.18) amounts to the computation of $\mathbf{z}^{(n+1)} := (\mathbf{u}^{(n+1)}, p^{(n+1)}, \mathbf{X}^{(n+1)})^T$, $0 \le n \le M-1$, as the solution of

the parameter dependent nonlinear system

$$\mathbf{G}\big(\mathbf{z}^{(n+1)};\tau_n\big)=0, \tag{1.19}$$

where the nonlinear mapping $\mathbf{G}(\cdot;\tau_n):\mathbb{R}^{\mathcal{N}}\to\mathbb{R}^{\mathcal{N}}$ is given by

$$\mathbf{G}(\mathbf{z};\tau_n)=\begin{pmatrix}(\mathbf{M}_1+\tau_n\mathbf{A})\mathbf{u}+\tau_n\mathbf{C}(\mathbf{u})+\tau_n\mathbf{B}^T p-\tau_n\mathbf{F}(\mathbf{X})-\mathbf{M}_1\mathbf{u}^{(n)}-\tau_n\mathbf{F}_0\\ \mathbf{B}\mathbf{u}-\mathbf{b}\\ \mathbf{M}_3\mathbf{X}-\mathbf{M}_3\mathbf{X}^{(n)}-\tau_n\mathbf{K}(\mathbf{X})\mathbf{u}\end{pmatrix}. \tag{1.20}$$

Here, $\mathbf{M}_1$, $\mathbf{A}$, and $\mathbf{B}$ are the mass and stiffness matrices associated with the fully discretized Navier–Stokes equations (1.17a), (1.17b). $\mathbf{M}_3$ is the mass matrix associated with (1.18). The nonlinear maps $\mathbf{C}(\mathbf{u})$ and $\mathbf{K}(\mathbf{X})$ are associated with the nonlinear parts of (1.17a), (1.18), respectively, and $\mathbf{b}$, $\mathbf{F}_0$ are vectors stemming from the inhomogeneous boundary data on Σ_{in}. For scaling purposes, we have multiplied the second block in (1.20) with τ_n^{-1}.

We note that the continuation parameter is the time t. We attempt to solve (1.19) by a path-following predictor–corrector continuation strategy with constant continuation as a predictor featuring an adaptive choice of the continuation steplength and a Newton-type method as a corrector [10]. A first result in this direction is the invertibility of the Jacobian $\mathbf{G}'(\mathbf{z};\tau_n)$.

Theorem 3 *For given $\mathbf{z}\in\mathbb{R}^{\mathcal{N}}$ and $\tau^{\min}>0$ there exists $\tau_n^{\max}(z)$ such that for all step sizes $\tau^{\min}\le\tau_n\le\tau_n^{\max}$ the Jacobian*

$$\mathbf{G}'(\mathbf{z};\tau_n)=\begin{pmatrix}\mathbf{M}_1+\tau_n\mathbf{A}+\tau_n\mathbf{C}'(\mathbf{u}) & \tau_n\mathbf{B}^T & -\tau_n\mathbf{F}'(\mathbf{X})\\ \mathbf{B} & \mathbf{0} & \mathbf{0}\\ -\tau_n\mathbf{K}(\mathbf{X}) & \mathbf{0} & \mathbf{M}_3-\tau_n\mathbf{K}'_{\mathbf{X}}(\mathbf{X},\mathbf{u})\end{pmatrix} \tag{1.21}$$

is invertible with bounded inverse

$$\big\|\big(\mathbf{G}'(\mathbf{z};\tau_n)\big)^{-1}\big\|\le\Lambda_n,$$

where Λ_n depends on $\tau^{\min}$ and $\tau_n^{\max}(z)$.

Proof For a proof we refer to [15]. □

The adaptive predictor-corrector continuation strategy is as follows:

Initialization Specify the initial variables $\mathbf{z}(t_0)=(\mathbf{u}^{(0)},p^{(0)},\mathbf{X}^{(0)})^T$, an initial continuation step size $\tau_{(0,0)}>0$, and bounds $\tau^{\min}$, $\tau^{\max}$, $\Theta_{\min}\ll 1$. Set $n=j=0$. Here, n is the iteration counter for the outer continuation, whereas j is the iteration counter for the inner predictor–corrector cycles.

Step 1: Predictor As long as $t_n<T$, set $t_{(n+1,j)}:=t_n+\tau_{(n,j)}$ and perform the continuation

$$\hat{\mathbf{z}}^{(0)}(t_{(n+1,j)}):=\mathbf{z}(t_n).$$

Step 2: Corrector Solve $\mathbf{G}(z;\tau_{(n,j)})=0$ by a combination of the ordinary and the simplified Newton method with initial guess $\hat{\mathbf{z}}^{(0)}(t_{(n+1,j)})$ and iteration counter $\ell=\ell(j)\geq 0$. Thereby, $\mathbf{z}$ gets updated by means of

$$\hat{\mathbf{z}}^{(\ell+1)}(t_{(n+1,j)}):=\hat{\mathbf{z}}^{(\ell)}(t_{(n+1,j)})+\alpha_\ell\Delta\mathbf{z}^{(\ell)},$$

where $\alpha_\ell>0$ is a suitable damping factor, and contraction factors Θ_ℓ, $\ell\geq 0$, are computed according to

$$\Theta_\ell:=\frac{\|\overline{\Delta\mathbf{z}}^{\ell+1}\|}{\|\overline{\Delta\mathbf{z}}^{\ell}\|},\quad \ell\geq 0.$$

The contraction factors serve as a convergence monitor in the simplified Newton method. If the simplified Newton corrector was successful, predict the new continuation step size by

$$\tau_{(n+1,0)}:=\min\left(\frac{(\sqrt{2}-1)\|\Delta\mathbf{z}^{(0)}\|}{2\max(\Theta_0,\Theta_{\min})\|\mathbf{z}(t_n)-\mathbf{z}(t_{n+1})\|}\tau_{(n,j)},\tau_{\max}\right),$$

where $t_{n+1}:=t_{(n+1,j)}$ and $\mathbf{z}(t_{n+1}):=\hat{\mathbf{z}}^{(\ell)}(t_{(n+1,j)})$. Set $n:=n+1$, $j:=0$ and go to Step 1.
Else correct the continuation step size τ by means of

$$\tau_{(n,j+1)}:=\frac{(\sqrt{2}-1)}{\sqrt{4\Theta_\ell+1}-1}\tau_{(n,j)}.$$

If $\tau_{(n,j+1)}<\tau_{\min}$, stop the algorithm (convergence failure). Otherwise, set $j:=j+1$ and go to Step 1.

1.5 Documentation of Numerical Results

In this section, we provide a documentation of simulation results for both the semi-explicit BE/FE FE-IB and the fully implicit BE/BE FE-IB.

1.5.1 The Semi-explicit BE/FE FE-IB

For studying the so-called tank treading motion [1, 12, 13, 16, 17] of viscoelastic particles under shear flow, we have applied the semi-explicit BE/FE FE-IB to a microchannel $\Omega=(0.0,1.2)\times(0.0,0.1)10^{-3}$ m with inflow boundary $\Gamma_{\text{in}}=\{0.0\}\times(0.0,0.1)$, inflow velocity $g=1.0\times 10^{-4}$ m/s, and outflow boundary $\Gamma_{\text{out}}:=\{1.2\}\times(0.0,0.1)$. We have further used $\rho=1.0\times 10^3$ kg/m^3 and $\mu=6.0\times 10^{-3}$ Pa/s resulting in a Reynolds number of approximately 0.1 which is

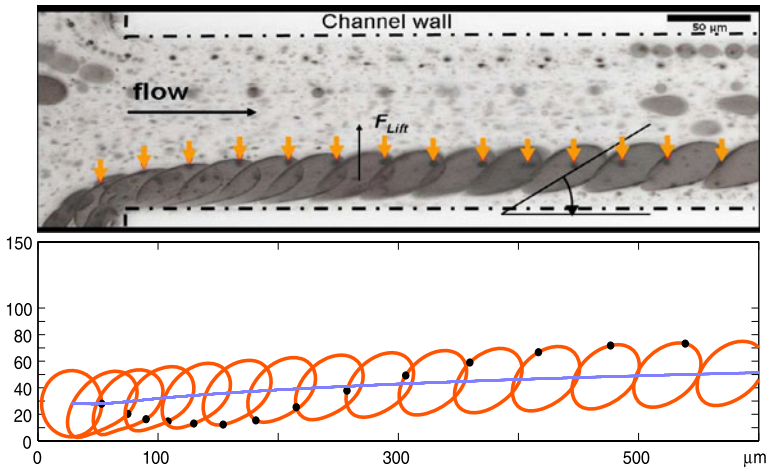


Fig. 1.2 Tank treading motion of a vesicle in shear flow: snapshots from experimental data (*top*) and numerical simulation by the FE/BE FE-IB (*bottom*)

typical for microfluidic flows. The elasticity coefficients have been chosen according to $\kappa_e = 6.0 \times 10^{-6}$ N/m and $\kappa_b = 2.0 \times 10^{-19}$ N/m, which correspond to the values of RBCs [11]. We have chosen a simplicial triangulation with $h = \sqrt{2}/16$ and an equidistant partitioning of $[0, L]$ with $\Delta q = h/2$ giving rise to a total of 42555 degrees of freedom. The time step has been chosen as $\Delta t = 1/240$.

Figure 1.2 displays snapshots from experimental data (top) and the simulation results due to the application of the BE/FE FE-IB (bottom). The initially spherical particle first gets deformed by attaining an ellipsoidal shape and then continues to move forward with a rotating membrane (cf. the motion of a material point on the immersed boundary marked by an arrow in the experimental result and by a black dot in the simulations). As can be observed as well, the particle experiences some lift towards the center of the channel. We computed an inclination angle of 34° which is in very good agreement with the experimental data.

The deformability of RBCs is such that they can pass through capillaries with diameters significantly less than the diameter of an RBC [27]. We have studied the motion of an RBC in a microchannel of width 20×10^{-6} m featuring a capillary of width 4.0×10^{-6} m (cf. Fig. 1.3). The inflow and outflow occur through the left and right boundary of the microchannel, respectively. The inflow velocity g as well as the other data ρ, μ, κ_e, κ_b have been chosen as in the previous example. For discretization in space and time, we have used $h = 1/15$ and $\Delta q = h/2$ yielding a total of 26709 degrees of freedom and a time step size of $\Delta t = 1/1000$. We note that Δt had to be chosen that small in order to satisfy the CFL condition (1.14), since the velocity at the opening of the capillary is almost five times higher than the inflow velocity. Figure 1.3 (top) displays the results obtained by the BE/FE FE-IB. For comparison, Fig. 1.3 (bottom) shows the snapshot of an RBC shortly before leaving a capillary in an experimental set up under essentially the same flow conditions. The

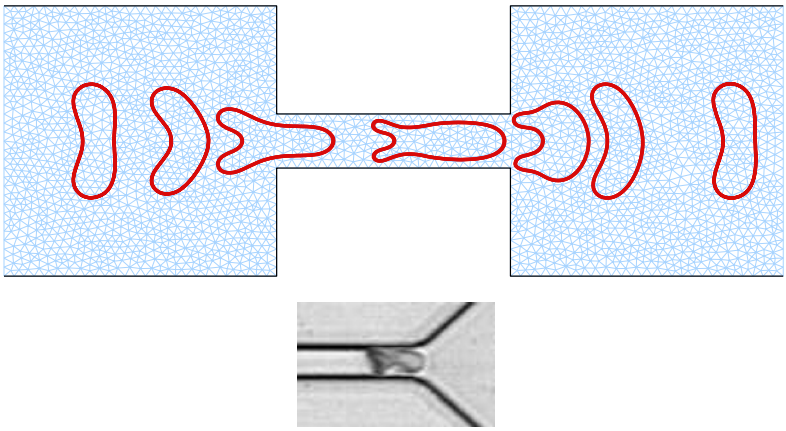


Fig. 1.3 RBC passing through a capillary of half its resting diameter: numerical simulation by the BE/FE FE-IB (*top*) and snapshot from an experiment (*bottom*)

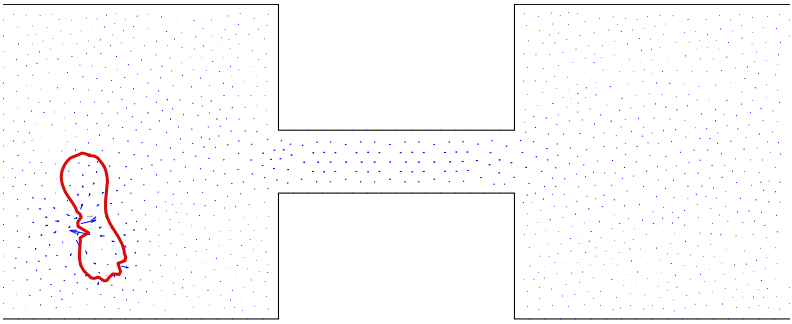


Fig. 1.4 Breakdown of the BE/FE FE-IB for $\tau = 1/250$ due to high oscillations of membrane nodes

fish-like shape of the RBC inside the capillary is very well captured by the numerical simulation.

1.5.2 The Fully Implicit BE/BE FE-IB

The motion of an RBC through a thin capillary is an appropriate example to illustrate the limitations of the semi-explicit BE/FE FE-IB and the advantages of the fully implicit BE/BE FE-IB. We have studied the same scenario as before, but applied the BE/FE FE-IB with a time step size $\Delta t = 1/250$. Figure 1.4 shows the onset of numerical instabilities due to oscillations of membrane nodes which caused a breakdown of the algorithm after $t = 0.05$. Such instabilities do not occur when

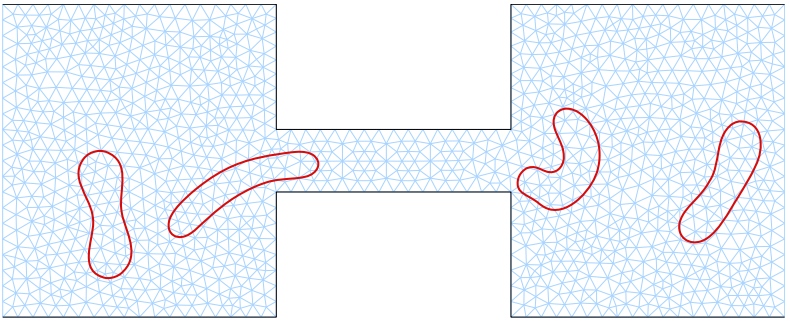


Fig. 1.5 Application of the BE/BE FE-IB: Snapshots of the RBC's membrane at selected time instants corresponding to the $*$-marked time instants in Fig. 1.6

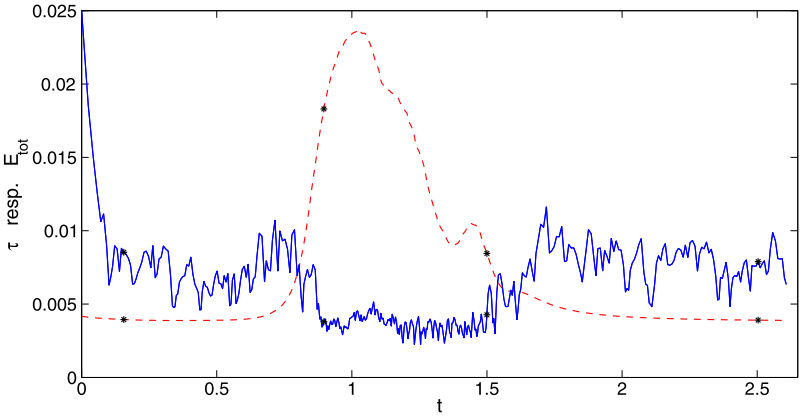


Fig. 1.6 Application of the BE/BE FE-IB: Evolution of the adaptively chosen time step sizes (*solid line*) and of the (*scaled*) total energy (*dashed line*)

using the fully implicit BE/BE FE-IB and its numerical realization by the predictor-corrector continuation strategy as described in Sect. 1.4 (cf. Fig. 1.5).

In fact, the adaptive time step size selection detects the critical stage of the process which occurs when the RBC starts to deform before entering the capillary and thus leads to a significant increase of its total energy. This is displayed in Fig. 1.6 which shows the evolution of the adaptively chosen time increments and the total energy.

Acknowledgements Both authors acknowledge support by the German National Science Foundation DFG within the DFG Priority Program SPP 1253 'Optimierung mit partiellen Differentialgleichungen'. The first author has been further supported by the NSF grants DMS-0707602, DMS-0914788, by the BMBF within the projects 'FROPT' and 'MeFreSim', and by the ESF within the Networking Programme 'OPTPDE'.

References

1. Abkarian M, Lartigue C, Viallat A (2002) Tank treading and unbinding of deformable vesicles in shear flow: determination of the lift force. Phys Rev Lett 88(6):068103
2. Anadere I, Chmiel H, Hess H, Thurston GB (1979) Clinical blood rheology. Biorheology 16(3):171–178
3. Boffi D, Gastaldi L (2003) A finite element approach for the immersed boundary method. Comput Struct 81(8–11):491–501
4. Boffi D, Gastaldi L, Heltai L (2007) Numerical stability of the finite element immersed boundary method. Math Models Methods Appl Sci 17(10):1479–1505
5. Brezzi F, Fortin M (1991) Mixed and hybrid finite element methods. Springer, New York
6. Chien S (1970) Shear dependence of effective cell volume as a determinant of blood viscosity. Science 168:977–979
7. Chien S (1987) Red cell deformability and its relevance to blood flow. Annu Rev Physiol 49:177–192
8. Chmiel H, Anadere I, Walitza E (1990) The determination of blood viscoelasticity in clinical hemorheology. Clinical Hemorheology 10:363–374
9. Cokelet GR (1980) Rheology and hemodynamics. Annu Rev Physiol 42:311–324
10. Deuflhard P (2004) Newton methods for nonlinear problems. Affine invariance and adaptive algorithms. Springer, Berlin
11. Eggleton CD, Popel AS (1998) Large deformation of red blood cell ghosts in simple shear flow. Phys Fluids 10:1834–1845
12. Fischer T, Schmid-Schönbein H (1977) Tank treading motion of red blood cell membranes in viscometric flow: behavior of intracellular and extracellular markers. Blood Cells 3:351–365
13. Fischer TM, Stöhr-Liesen M, Schmid-Schönbein H (1978) The red cell as a fluid droplet—tank-treading like motion of the human erythrocyte membrane in shear flow. Science 202:894–896
14. Franke T, Hoppe RHW, Linsenmann C, Schmid L, Willbold C, Wixforth A (2011) Numerical simulation of the motion and deformation of red blood cells and vesicles in microfluidic flows. Comput Vis Sci 14(4):167–180
15. Hoppe RHW, Linsenmann C (2011) An adaptive Newton continuation strategy for the fully implicit finite element immersed boundary method. Submitted to J Comp Phys
16. Kantsler V, Steinberg V (2005) Orientation and dynamics of a vesicle in tank-treading motion in shear flow. Phys Rev Lett 95:258101
17. Keller SR, Skalak R (1982) Motion of a tank-treading ellipsoidal particle in a shear flow. J Fluid Mech 120:27–47
18. Le DV, White J, Peraire J, Lim KM, Khoo BC (2009) An implicit immersed boundary method for three-dimensional fluid-membrane interactions. J Comput Phys 228(22):8427–8445
19. Lee P, Griffith BE, Peskin CS (2010) The immersed boundary method for advection-electrodiffusion with implicit timestepping and local mesh refinement. J Comput Phys 229(13):5208–5227
20. Mori Y, Peskin CS (2008) Implicit second-order immersed boundary methods with boundary mass. Comput Methods Appl Mech Eng 197(25–28):2049–2067
21. Newren EP, Fogelson AL, Guy RD, Kirby RM (2007) Unconditionally stable discretizations of the immersed boundary equations. J Comp Physiol 222(2):702–719
22. Noguchi H, Gompper G (2004) Fluid vesicles with viscous membranes in shear flow. Phys Rev Lett 93:258102
23. Pan T-W, Wang T (2009) Dynamical simulation of red blood cell rheology in microvessels. Int J Numer Anal Model 6(3):455–473
24. Peskin CS (1977) Numerical analysis of blood flow in the heart. J Comput Phys 25(3):220–252
25. Peskin CS (2002) The immersed boundary method. Acta Numer 11:479–517
26. Pozrikidis C (2003) Modeling and simulation of capsules and biological cells. Chapman & Hall/CRC, Boca Raton

27. Pozrikidis C (2005) Axisymmetric motion of a file of red blood cells through capillaries. Phys Fluids 17(3):031503
28. Stockie JM, Wetton BR (1999) Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes. J Comput Phys 154(1):41–64
29. Tartar L (2007) An introduction to Sobolev spaces and interpolation spaces. Springer, Berlin
30. Thurston GB (1972) Viscoelasticity of human blood. Biophys J 12(9):1205–1217
31. Thurston GB (1996) Viscoelastic properties of blood and blood analogs. In: How TV (ed) Advances in hemodynamics and hemorheology. JAI Press, London, pp 1–30
32. Tu C, Peskin CS (1992) Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods. SIAM J Sci Stat Comput 13(6):1361–1376
33. Wang T, Pan T-W, Xing ZW, Glowinski R (2009) Numerical simulation of red blood cell rouleaus in microchannels. Phys Rev E 79(4):041916-1

Chapter 2
Iterative Solution Methods for Large-Scale Constrained Saddle-Point Problems

Erkki Laitinen and Alexander Lapin

Abstract Iterative solution methods for a class of finite-dimensional constrained saddle point problems are developed. These problems arise if variational inequalities and minimization problems are solved with the help of mixed finite element statements involving primal and dual variables. In the paper, we suggest several new approaches to the construction of saddle point problems and present convergence results for the iteration methods. Numerical results confirm the theoretical analysis.

Keywords Variational inequality · Optimal control problem · Finite element method · Constrained saddle point problem · Iteration methods

2.1 Introduction

We construct and investigate iteration methods for the finite dimensional constrained saddle point problem

$$\begin{pmatrix} A & -C^T \\ -C & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} + \begin{pmatrix} P(x) \\ -Q(\lambda) \end{pmatrix} \ni \begin{pmatrix} f \\ -g \end{pmatrix}, \tag{2.1}$$

where $f \in \mathbb{R}^{N_x}$ and $g \in \mathbb{R}^{N_\lambda}$ are given vectors, and the following assumptions hold:

(A1) Operator $A : \mathbb{R}^{N_x} \to \mathbb{R}^{N_x}$ is continuous, strictly monotone and coercive;
(A2) $C \in \mathbb{R}^{N_\lambda \times N_x}$, $N_\lambda \le N_x$, is a full rank matrix: $\operatorname{rank} C = N_\lambda$;

E. Laitinen (✉)
University of Oulu, P.O. Box 3000, 90014, Oulu, Finland
e-mail: erkki.laitinen@oulu.fi

A. Lapin
Kazan Federal University, Kremljovskaja St., Kazan 420008, Russia
e-mail: alapin@ksu.ru

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_2,

(A3) $P = \partial\Phi$, $Q = \partial\Psi$, where $\Phi : \mathbb{R}^{N_x} \to \bar{\mathbb{R}}$ and $\Psi : \mathbb{R}^{N_\lambda} \to \bar{\mathbb{R}}$ are proper, convex and lower semi-continuous functions.

Different particular cases of the problem (2.1) arise if grid approximations (finite difference, finite element, etc.) are used to approximate variational inequalities or optimal control problems. Specifically, introducing the dual variables to the grid approximations of the variational inequalities with constraints for the gradient of a solution leads to (2.1) with $Q = 0$. Approximations of the control problems with control function in the right-hand side of a linear differential equation or in the boundary conditions give rise to the saddle point problem (2.1) with $Q = 0$ and linear A. Finally, we note that mixed and hybrid finite element schemes for the 2-nd order variational inequalities with pointwise constraints to the solution imply (2.1) with $P = 0$.

The solution methods for large-scale unconstrained saddle point problems are thoroughly investigated. The state-of-the-art for this problem can be found in the survey paper [1] and in the book [6]. Constrained saddle point problems arising from the Lagrangian approach for solving variational inequalities in mechanics and physics are considered in [8–10] (see also the bibliography therein). Namely, the convergence of Uzawa-type, Arrow-Hurwitz-type, and operator-splitting iterative methods are investigated in these books.

The development of the efficient numerical methods designed to solve state-constrained optimal control problems represents severe numerical challenges. The construction of the effective iterative solution methods for them is an actual problem. The achievements in this field during the past two decades are reported in the book [5] and the articles [2–4, 11–15, 21]. The augmented Lagrangian method as well as regularization and penalty methods have been investigated for particular classes of the state-constrained optimal control problems. Adjustment schemes for the regularization parameter of a Moreau–Yosida-based regularization and for the relaxation parameter of interior point approaches to the numerical solution of pointwise state constrained elliptic optimal control problems have been constructed. Lavrentiev regularization has been applied to transform the state constraints to the mixed control-state constraints in the linear-quadratic elliptic control problem with pointwise constraints on the state. The interior point methods and the primal-dual active set strategy have been applied to the transformed problem.

In this article, we prove convergence of the iterative solution methods for the saddle point problem (2.1). The sufficient conditions of convergence for the iterative methods are presented in the form of matrix inequalities and give rise to constructing appropriate preconditioners and allow choosing the iterative parameter. Applications of the general convergence results to sample examples of the variational inequalities and optimal control problems, as well as several numerical results, are included. The results of this article are founded in the previous papers [16–19] by the authors.

2.2 Iterative Methods for the Saddle-Point Problem

2.2.1 Existence of the Solutions

Consider the problem (2.1) and suppose that it has a nonempty set of solutions $X = \{(x, \lambda)\}$. Below we present the existence results for the cases $P = 0$ or $Q = 0$, which are mostly interesting for the applications included in the article. Note that the assumptions (A1)–(A3) ensure the uniqueness of the component x.

Lemma 2.1 *Let the assumptions* (A1)–(A3) *be fulfilled and* $P = 0$. *Let also the operator* A *be uniformly monotone, i.e.,*

$$(Ax - Ay, x - y) \geq \alpha \|x - y\|_{A_0}^2 \quad \alpha > 0, \tag{2.2}$$

and Lipshitz-continuous

$$\|Ax - Ay\|_{A_0^{-1}} \leq \beta \|x - y\|_{A_0} \tag{2.3}$$

with a symmetric and positive definite matrix $A_0 \in \mathbb{R}^{N_x \times N_x}$. *Then, the problem* (2.1) *has a unique solution* (x, λ).

Lemma 2.2 ([17]) *Let the assumptions* (A1)–(A3) *be fulfilled,* $Q = 0$, *and*

$$\operatorname{int} \operatorname{dom} \Phi \cap \left\{x \in \mathbb{R}^{N_x} : Cx = g\right\} \neq \emptyset.$$

Then, the problem (2.1) *has a nonempty set of solutions* $X = \{(x, \lambda)\}$ *with a uniquely defined component* x.

2.2.2 Iteration Methods

We consider two iteration methods for solving (2.1): a preconditioned Uzawa-type method

$$\begin{aligned} Ax^{k+1} + P\left(x^{k+1}\right) - C^T \lambda^k &\ni f, \\ \frac{1}{\tau} B_\lambda \left(\lambda^{k+1} - \lambda^k\right) + Q\left(\lambda^{k+1}\right) + Cx^{k+1} &\ni g \end{aligned} \tag{2.4}$$

and a preconditioned Arrow-Hurwitz-type method

$$\begin{aligned} \frac{1}{\tau} B_x \left(x^{k+1} - x^k\right) + Ax^k + P\left(x^{k+1}\right) - C^T \lambda^k &\ni f, \\ \frac{1}{\tau} B_\lambda \left(\lambda^{k+1} - \lambda^k\right) + Q\left(\lambda^{k+1}\right) + Cx^{k+1} &\ni g. \end{aligned} \tag{2.5}$$

Preconditioners B_x and B_λ are supposed to be symmetric and positive definite matrices, $\tau > 0$ is an iteration parameter.

In the forthcoming theorem, we give sufficient conditions of the convergence for the iterative method (2.4).

Theorem 2.1 ([17]) *Let the operator A be uniformly monotone (2.2). If*

$$B_\lambda > \frac{\tau}{2\alpha} C A_0^{-1} C^T, \tag{2.6}$$

then the iterations of the method (2.4) converge to a solution of (2.1) starting from any initial guess λ^0.

Note 1 Since the component x of the exact solution (x, λ), as well as the components x^k of the iterations belong to $D(P) \subset \operatorname{dom} \Phi$, it is sufficient for A to be a uniform monotone operator only on $\operatorname{dom} \Phi$.

Note 2

(a) In [6], it is proved that the positive eigenvalues μ of two generalized eigenvalue problems

$$C A_0^{-1} C^T = \mu B_\lambda \quad \text{and} \quad C^T B_\lambda^{-1} C = \mu A_0$$

with symmetric and positive definite matrices A_0 and B_λ coincide. Owing to this inequality, (2.6) is equivalent to the inequality

$$A_0 > \frac{\tau}{2\alpha} C^T B_\lambda^{-1} C. \tag{2.7}$$

(b) The inequality

$$(Ax - Ay, x - y) > \frac{\tau}{2}\big(C^T B_\lambda^{-1} C(x - y), x - y\big) \quad \forall x \neq y$$

replaces both (2.2) and (2.6).

(c) If A is linear then we can take $A_0 = 0/5(A + A^T)$ and the inequalities (2.6) and (2.7) become, respectively (cf. [18]):

$$B_\lambda > \frac{\tau}{2} C A_0^{-1} C^T \quad \text{and} \quad A_0 > \frac{\tau}{2} C^T B_\lambda^{-1} C.$$

(d) In the case of a potential operator $A : A = \nabla \Xi$, where Ξ is a differentiable convex function, the method (2.4) is just the preconditioned Uzawa method applied to finding a saddle point of the Lagrangian

$$2\mathscr{L}(x, \lambda) = \frac{1}{2}\Xi(x) + \Phi(x) - (\lambda, Cx - g) - (f, x).$$

The sufficient conditions for the choice of the preconditioning matrices B_x and B_λ and iterative parameter $\tau > 0$ required to ensure the convergence of the Arrow–Hurwitz-type method (2.5) are given by

Theorem 2.2 ([17]) *Let the operator A be uniformly monotone* (2.2) *and Lipshitz-continuous* (2.3). *If*

$$\left(2\alpha - \tau\mu_{\max}\beta^2\right)A_0 > \tau\, C^T B_\lambda^{-1} C, \tag{2.8}$$

where $\mu_{\max} = \lambda_{\max}(B_x^{-1/2} A_0 B_x^{-1/2})$ is the maximal eigenvalue of the matrix $B_x^{-1/2} A_0 B_x^{-1/2}$, then iterations of the method (2.5) *converge to a solution of* (2.1) *starting from any initial guess (x^0, λ^0).*

Note 3 It is sufficient for A to be a uniform monotone and Lipshitz-continuous operator only on $\operatorname{dom}\Phi$ (cf. Note 1).

Note 4

(a) The choice $B_x = A_0$ gives the best limit for the iterative parameter τ ensuring the convergence of the method. In this case, the inequality (2.8) reads

$$A_0 > \frac{\tau}{2\alpha - \tau\beta^2}\, C^T B_\lambda^{-1} C,$$

and further choice of a preconditioner B_λ is similar to the case of the method (2.4).

(b) If A is linear then the sufficient convergence condition (2.8) can be replaced by the following sharper condition:

$$A > \frac{\tau}{2}\left(A B_x^{-1} A^T + C^T B_\lambda^{-1} C\right).$$

2.2.3 Stopping Criterion

One possible stopping criterium for an iterative process is based on the evaluation of residual norms. Namely, when solving the problem (2.1) by an iterative method we find not only the pair (x^k, λ^k)—approximations of the exact solution (x, λ), but also uniquely defined selections $\gamma^k \in P(x^k)$, $\delta^k \in Q(\lambda^k)$. Let us define the residual vectors

$$r_x^k = f - Ax^k - \gamma^k + C^T\lambda^k, \qquad r_\lambda^k = g - \delta^k - Cx^k.$$

Lemma 2.3 *Let the operator A be uniformly monotone* (2.2). *Then the error estimate*

$$\left\|x - x^k\right\|_{A_0} \le c_1 \left\|r_x^k\right\|_{A_0^{-1}} + c_2 \left\|\lambda - \lambda^k\right\|^{1/2} \left\|r_\lambda^k\right\|^{1/2} \quad \forall k \tag{2.9}$$

is valid for the methods (2.4) *and* (2.5). *Constants c_1 and c_2 depend only on the constant α of uniform monotonicity of operator A.*

Since $\|\lambda - \lambda^k\| \to 0$ for $k \to \infty$, then the inequality (2.9) gives an estimate for the error $\|x - x^k\|_{A_0}$ throughout the norms $\|r_x^k\|_{A_0^{-1}}$ and $\|r_\lambda^k\|$.

Note 5 In the Uzawa-type method for the saddle point problem, the inclusion $Ax - B^T\lambda + \partial\varphi(x) \ni f$ is solved exactly on each iteration. Due to this fact, $r_x^k = 0$ and the estimate (2.9) reads

$$\left\|x - x^k\right\|_{A_0} \le c_2 \left\|\lambda - \lambda^k\right\|^{1/2} \left\|r_\lambda^k\right\|^{1/2} \quad \forall k, \tag{2.10}$$

whence

$$\left\|x - x^k\right\| = o\left(\left\|r_\lambda^k\right\|^{1/2}\right) \quad \text{for } k \to \infty.$$

2.3 Application to Variational Inequalities

Now we consider the application of the previous results to a sample example of the variational inequality: find $u \in V$ such that $\forall v \in V$

$$\int_\Omega a(x)\, k(\nabla u) \cdot \nabla(v-u)\,\mathrm{d}x + \int_\Omega |\nabla v|\,\mathrm{d}x - \int_\Omega |\nabla u|\,\mathrm{d}x \ge \int_\Omega f(v-u)\,\mathrm{d}x. \tag{2.11}$$

Here $H_0^1(\Omega) \subset V \subset H^1(\Omega)$, $a(x) > 0$, and $k(\bar{t}) : \mathbb{R}^2 \to \mathbb{R}^2$ is a continuous and uniformly monotone vector-function:

$$\big(k(\bar{t}_1) - k(\bar{t}_2)\big) \cdot (\bar{t}_1 - \bar{t}_2) \ge \sigma_0 |\bar{t}_1 - \bar{t}_2|^2 \quad \forall \bar{t}_i,\ \sigma_0 > 0. \tag{2.12}$$

We construct a simple finite element approximation of (2.11) in the case of polygonal domain Ω. Let $\overline{\Omega} = \bigcup_{e \in T_h} e$ be a conforming triangulation of $\overline{\Omega}$ [7], where T_h is a family of N_e non-overlapping closed triangles e (finite elements) and h is the maximal diameter of all $e \in T_h$. Further $V_h \subset H_0^1(\Omega)$ is the space of the continuous and piecewise linear functions (linear on each $e \in T_h$), while $U_h \in L_2(\Omega)$ is the space of the piecewise constant functions. Define $f_h \in U_h$ and $a_h \in U_h$ by the equalities

$$f_h(x) = |e|^{-1} \int_{t \in e} f(t)\,\mathrm{d}t, \qquad a_h(x) = |e|^{-1} \int_{t \in e} a(t)\,\mathrm{d}t, \quad \forall x \in e,\ |e| = \operatorname{meas} e.$$

The finite element approximation of the problem (2.11) satisfies the relation

$$u_h \in V_h : \int_\Omega a_h(x) k(\nabla u_h) \cdot \nabla(v_h - u_h)\,\mathrm{d}x + \int_\Omega |\nabla v_h|\,\mathrm{d}x - \int_\Omega |\nabla u_h|\,\mathrm{d}x$$
$$\ge \int_\Omega f_h(v_h - u_h)\,\mathrm{d}x, \quad \forall v_h \in V_h. \tag{2.13}$$

In order to formulate (2.13) in a vector-matrix form, we first define the vectors $u \in \mathbb{R}^{N_u}$ and $w \in \mathbb{R}^{N_e}$ of the nodal values of functions $u_h \in V_h$ and $w_h \in U_h$, respectively. We correspond a vector valued function $\bar{q}_h = (q_{1h}, q_{2h}) \in U_h \times U_h$ to the vector $q = (q_{11}, q_{21}, \ldots, q_{1i}, q_{2i}, \ldots, q_{1N_e}, q_{2N_e}) \in \mathbb{R}^{2N_y}$, where $q_{1i} = q_{1h}(x)$, $q_{2i} = q_{2h}(x)$ for $x \in e_i$. Further, we define the matrix $L \in \mathbb{R}^{N_u \times N_y}$ and the operator $k : \mathbb{R}^{N_y} \to \mathbb{R}^{N_y}$ by the equalities

$$(Lu, q) = \int_\Omega \nabla u_h(x) \cdot \bar{q}_h(x)\,\mathrm{d}x, \qquad \big(k(p), q\big) = \int_\Omega a_h(x) k\big(\bar{p}_h(x)\big) \cdot \bar{q}_h(x)\,\mathrm{d}x,$$

diagonal matrix $D = \mathrm{diag}(a_1, a_1, \ldots, a_i, a_i, \ldots, a_{N_e}, a_{N_e}) \in \mathbb{R}^{N_y \times N_y}$ with the entries $a_i = a_h(x)$ for $x \in e_i$, and vector $f \in \mathbb{R}^{N_u}$, $(f, u) = \int_\Omega f_h(x) u_h(x)\,\mathrm{d}x$. Finally, let the convex function be defined by the relation

$$\theta(p) = \sum_{j=1}^{N_e} |e_j| \left(p_{2j}^2 + p_{2j-1}^2\right)^{1/2}.$$

Now, the discrete variational inequality (2.13) can be written in the form

$$u \in \mathbb{R}^{N_u} : \big(D\,k(Lu), L(v-u)\big) + \theta(Lv) - \theta(Lu) \geq (f, v-u) \quad \forall v \in \mathbb{R}^{N_u}$$

or, equivalently, as the inclusion

$$L^T D k(Lu) + L^T \partial\theta(Lu) \ni f. \tag{2.14}$$

We will construct different saddle point problems using the inclusion (2.14).

2.3.1 Variational Inequality with the Linear Main Operator

First, let us consider the discrete problem approximating variational inequality with the linear differential operator: $k(\nabla u) = \nabla u$. The corresponding discrete inclusion is

$$L^T\, D\, Lu + L^T \partial\theta(Lu) \ni f.$$

Denoting $p = Lu$, we transform it to one of the following three systems:

$$\frac{1}{2} L^T D L u + L^T \lambda = f, \quad \lambda \in \frac{1}{2} D p + \partial\theta(p), \quad p = Lu; \tag{2.15}$$

$$L^T D L u + L^T \lambda = f, \quad \lambda \in \partial\theta(p), \quad p = Lu; \tag{2.16}$$

$$L^T \lambda = f, \quad \lambda \in D p + \partial\theta(p), \quad p = Lu. \tag{2.17}$$

The matrix $A_1 = \begin{pmatrix} 0.5 L^T D L & 0 \\ 0 & 0.5 D \end{pmatrix}$ of the first two equations in the system (2.15) is positive definite and block diagonal. Thus, the Uzawa-type method (2.4), being

applied to this system, can be effectively implemented. On the other side, the saddle point problems (2.16) and (2.17) contain degenerate matrices $A_2 = \left(\begin{smallmatrix} L^T DL & 0 \\ 0 & 0 \end{smallmatrix}\right)$ and $A_3 = \left(\begin{smallmatrix} 0 & 0 \\ 0 & D \end{smallmatrix}\right)$, respectively, so, the iterative method (2.4) cannot be applied for their solution. We realize different equivalent transformations of (2.16) and (2.17) by using the equation $Lu = p$, to obtain the systems with positive definite matrices A_i. In particular, we can get the system corresponding to the augmented Lagrangian method

$$\begin{pmatrix} (1+r)L^T DL & -rL^T D & L^T \\ -rDL & rD & -E \\ L & -E & 0 \end{pmatrix} \begin{pmatrix} u \\ p \\ \lambda \end{pmatrix} + \begin{pmatrix} -f \\ \partial\theta(p) \\ 0 \end{pmatrix} \ni 0, \quad r > 0. \tag{2.18}$$

The matrix $A_r = \left(\begin{smallmatrix} (1+r)L^T DL & -rL^T D \\ -rDL & rD \end{smallmatrix}\right)$ in (2.18) is symmetric and positive definite for any $r > 0$. However, it is not block diagonal or block triangle. In view of this, the method (2.4) cannot be effectively implemented (while it converges for this problem). The most well-known methods for solving (2.18) are the so-called Algorithms 2–6 (see [8, 9]), based on the block relaxation technique to inverse A_r and updating of the Lagrange multipliers λ. Instead of (2.18) we construct the systems with positive definite and block triangle 2×2 left upper blocks:

$$\begin{pmatrix} L^T DL & 0 & L^T \\ -rDL & rD & -E \\ L & -E & 0 \end{pmatrix} \begin{pmatrix} u \\ p \\ \lambda \end{pmatrix} + \begin{pmatrix} -f \\ \partial\theta(p) \\ 0 \end{pmatrix} \ni 0, \tag{2.19}$$

$$\begin{pmatrix} rL^T DL & -rL^T D & L^T \\ 0 & D & -E \\ L & -E & 0 \end{pmatrix} \begin{pmatrix} u \\ p \\ \lambda \end{pmatrix} + \begin{pmatrix} -f \\ \partial\theta(p) \\ 0 \end{pmatrix} \ni 0. \tag{2.20}$$

Lemma 2.4 *Let* $0 < r < 4$. *Then the matrices*

$$A_2[r] = \begin{pmatrix} L^T DL & 0 \\ -rDL & rD \end{pmatrix}, \qquad A_3[r] = \begin{pmatrix} rL^T DL & -rL^T D \\ 0 & D \end{pmatrix} \tag{2.21}$$

in the systems (2.19) *and* (2.20) *are energy equivalent to the block diagonal and positive definite matrix*

$$A_0 = \begin{pmatrix} L^T DL & 0 \\ 0 & D \end{pmatrix}$$

with the constants depending only on r:

$$\alpha_i(r)(A_0 x, x) \le \big(A_i[r]x, x\big) \le \beta_i(r)(A_0 x, x) \quad \forall x,\ i = 2, 3.$$

As the matrices $A_2[r]$ and $A_3[r]$ defined in (2.21) are block triangle, the Uzawa-type iterative method (2.4) can be easily implemented for the solution of the systems (2.19) and (2.20). Owing to Theorem 2.1, the most reasonable preconditioner is $B_\lambda = D^{-1}$. The convergence result in the particular case $r = 1$ reads as follows:

Theorem 2.3 ([18]) *Let $r = 1$. Then the method (2.4) with $B_\lambda = D^{-1}$ applied to the systems (2.19) and (2.20) converges provided that $0 < \tau < \frac{1}{2}$.*

Implementation of the method (2.4) for (2.19) and (2.20) includes solving a system of linear equations with the matrix $L^T DL$ and solving an inclusion of the form $cDp + \partial\theta(p) \ni F$, $c = \text{const}$ with a known vector F. In the example under consideration, the matrix D is diagonal and the multivalued operator $\partial\theta$ is block-diagonal with 2×2 blocks. Because of this, the inclusion $cDp + \partial\theta(p) \ni F$ can be easily solved by the direct methods.

2.3.2 Variational Inequality with Non-linear Main Operator

To construct saddle point problems for the inclusion (2.14) with the non-linear main operator, we proceed similarly to the linear case. Namely, by using Lagrange multipliers λ and the equation $Lu = p$, we construct saddle point problems with uniformly monotone operators in the space of the vectors $x = (u, p)^T$. Consider two of them:

$$\begin{aligned} &L^T k(Lu) + L^T\lambda = 0, \qquad -rDLu + rDp + \partial\theta(p) - \lambda \ni 0, \\ &\quad Lu - p = 0, \end{aligned} \tag{2.22}$$

$$\begin{aligned} &rL^T DLu - rL^T Dp + L^T\lambda = 0, \qquad Dk(p) + \partial\theta(p) - \lambda \ni 0, \\ &\quad Lu - p = 0. \end{aligned} \tag{2.23}$$

The systems (2.22) and (2.23) contain block-triangle operators

$$A_1(x) = \begin{pmatrix} L^T k(Lu) & 0 \\ -rDLu & rDp \end{pmatrix} \quad \text{and} \quad A_2(x) = \begin{pmatrix} rL^T DLu & -rL^T Dp \\ 0 & Dk(p) \end{pmatrix}.$$

Lemma 2.5 *Let the uniform monotonicity property (2.12) with the constant σ_0 hold and $0 < r < 4\sigma_0$. Then the operators A_1 and A_2 are uniformly monotone*:

$$(A_i x_1 - A_i x_2, x_1 - x_2) \geq \alpha_i \|x_1 - x_2\|^2_{A_0}, \quad \alpha_i = \alpha_i(r, \sigma_0) > 0, \ i = 1, 2, \tag{2.24}$$

where $A_0 = \begin{pmatrix} L^T DL & 0 \\ 0 & D \end{pmatrix}$ is the positive definite matrix.

Lemma 2.6 *Let the function k be Lipschitz-continuous*:

$$\big(k(\bar{t}_1) - k(\bar{t}_2)\big) \cdot (\bar{s}) \leq \sigma_1 |\bar{t}_1 - \bar{t}_2||\bar{s}| \quad \forall \bar{t}_i, \bar{s}. \tag{2.25}$$

Then the operators A_1 and A_2 are Lipschitz-continuous:

$$\|A_i x_1 - A_i x_2\|_{A_0^{-1}} \leq \beta_i \|x_1 - x_2\|_{A_0}, \quad \beta_i = \beta_i(r, \sigma_1), \ i = 1, 2. \tag{2.26}$$

Application of Lemmas 2.5 and 2.6 and Theorem 2.1 gives the following result:

Theorem 2.4 *Let* $0 < r < 4\sigma_0$. *Then the Uzawa-type iterative method* (2.4) *with the preconditioner* $B_\lambda = D^{-1}$ *applied for solving* (2.22) *and* (2.23) *converges if*

$$0 < \tau < \frac{2\alpha_2 r}{1+r}.$$

Implementation of the method (2.4) for (2.23) includes solving a system of linear equations with the matrix $L^T DL$ and solving the inclusion $Dk(p) + \partial\theta(p) \ni F$ with a known vector F. This inclusion can be effectively solved because the operator k is diagonal and $\partial\theta$ is a 2×2 block diagonal operator.

Implementation of (2.4) for the problem (2.22) requires solving the system of nonlinear equations $L^T k(Lu) + L^T \lambda = 0$ by an inner iterative method. Thus, the effectiveness of the algorithm depends also on the effectiveness of an inner iterative method. Instead of the Uzawa-type method we can apply the Arrow–Hurwitz-type iterative method (2.5) for the problem (2.22) with $B_\lambda = D^{-1}$ and $B_x = A_0 = \left(\begin{smallmatrix} L^T DL & 0 \\ 0 & D \end{smallmatrix}\right)$. The results of Lemmas 2.5 and 2.6 and Theorem 2.2 yield

Theorem 2.5 *Let* $0 < r < 4\sigma_0$. *Then the iterative method* (2.5) *for the problem* (2.22)

$$\begin{aligned} \frac{r}{\tau} L^T DL\big(u^{k+1} - u^k\big) + L^T k\big(Lu^k\big) + L^T \lambda^k &= 0, \\ \frac{1}{\tau} D\big(p^{k+1} - p^k\big) - r\,DLu^k + r\,Dp^k + \partial\theta\big(p^{k+1}\big) - \lambda^k &\ni 0, \\ \frac{1}{\tau}\big(\lambda^{k+1} - \lambda^k\big) + D\big(Lu^{k+1} - p^{k+1}\big) &= 0 \end{aligned} \tag{2.27}$$

converges if

$$\tau < \frac{2\alpha_1}{\beta_1 + (1+r)/r}.$$

It is easy to see that the implementation of (2.27) includes the same steps as the implementation of the method (2.4) for (2.23).

2.3.3 Variational Inequality with Pointwise Constraints both for the Solution and Its Gradient

Consider the variational inequality: find $u \in U_{ad} = \{u \in H_0^1(\Omega) : u(x) \geq 0 \text{ in } \Omega\}$, such that for all $v \in U_{ad}$

$$\int_\Omega a(x) k\big(|\nabla u|\big) \nabla u \cdot \nabla(v-u)\,\mathrm{d}x + \int_\Omega \big(|\nabla v| - |\nabla u|\big)\,\mathrm{d}x \geq \int_\Omega f(v-u)\,\mathrm{d}x,$$

where $a(x) > 0$ and the vector-function $k(|\bar{t}|)\bar{t}$ satisfies (2.12). After approximation of this variational inequality, we obtain the discrete variational inequality

$$\big(D\,k(Lu), L(v-u)\big) + \theta(Lv) - \theta(Lu) + \varphi(v) - \varphi(u) \geq (f, v-u) \quad \forall v \in \mathbb{R}^{N_u},$$

where φ is the indicator function of the constraint set $\{u \in \mathbb{R}^{N_u} : u_i \geq 0\ \forall i\}$, while all other notations are the same as above. We write this variational inequality in the form of inclusion

$$L^T\,D\,k(Lu) + L^T\partial\theta(Lu) + \partial\varphi(u) \ni f.$$

We proceed as before and construct the saddle point problems

$$L^T k(Lu) + \partial\varphi(u) + L^T\lambda = 0, \qquad -r\,DLu + r\,Dp + \partial\theta(p) - \lambda \ni 0,$$
$$Lu - p = 0, \tag{2.28}$$
$$rL^T\,DLu - rL^T\,Dp + \partial\varphi(u) + L^T\lambda = 0, \qquad Dk(p) + \partial\theta(p) - \lambda \ni 0,$$
$$Lu - p = 0. \tag{2.29}$$

Both iterative methods, (2.4) and (2.5), can be applied for solving these saddle point problems because the results of Theorems 2.1 and 2.2 are valid with the operator P defined by $P(x) = (\partial\varphi(u), \partial\theta(p))^T$. But now, the implementation of the Uzawa-type iterative method (2.4) for (2.29) includes the solution of the finite dimensional obstacle problem—the inclusion

$$rL^T\,DLu + \partial\varphi(u) \ni rL^T\,Dp - L^T\lambda$$

with the symmetric and positive definite matrix $rL^T\,DL$, and the implementation of this method for (2.28) includes the solution of the problem with the non-linear operator

$$L^T k(Lu) + \partial\varphi(u) \ni -L^T\lambda.$$

The Arrow–Hurwitz-type method (2.5) with preconditioners $B_x = \left(\begin{smallmatrix} D & 0 \\ 0 & D \end{smallmatrix}\right)$ and $B_\lambda = D^{-1}$ being applied to (2.28) or (2.29) converges and it can be easily implemented. On the other hand, in this case the maximal eigenvalue $\mu_{\max}$ of the matrix $B_x^{-1/2} A_0 B_x^{-1/2}$ depends on condition numbers of the matrices D and $L^T L$, thus, on the mesh step h. Convergence of the corresponding iterative methods is guaranteed for the very small iterative parameter τ, and numerical experiments demonstrate slow convergence of the Arrow–Hurwitz-type method (2.5).

2.3.4 Results of Numerical Experiments

We have solved a number of 1D and 2D linear and non-linear variational inequalities using the simplest finite element and finite difference approximations and applying

Table 2.1 Dependence of n_{it} on τ and n for Problem 2.1

n	5000					50000	500000
τ	1.3	1.2	1.1	1	0.9	1	1
n_{it}	10	8	6	2	6	2	2

the Uzawa-type method. The main purpose of the numerical experiments was to observe the dependence of the number of iterations upon the mesh step h and iterative parameter τ. We also compared proposed iterative algorithms with well-known algorithms for saddle point problems constructed via an augmented Lagrangian technique. Several numerical results are reported below.

Consider the following one-dimensional variational inequality

$$u \in K : \int_0^1 u'\big(v' - v'\big)\,\mathrm{d}x \geq \int_0^1 f(v-u)\,\mathrm{d}x \quad \forall v \in K$$

with the set of constraints $K = \{u \in H_0^1(0,1) : |u'(x)| \leq 1 \text{ for } x \in (0,1)\}$. Finite element approximation with piecewise linear elements on the uniform grid leads to the inclusion $L^T Lu + L^T \partial\theta(Lu) \ni f$, where the matrix L corresponds to the approximation of the first order derivative. We solve the corresponding saddle point problems:

Problem 2.1 The saddle point problem with $A = \left(\begin{smallmatrix} L^T L & 0 \\ -L & E \end{smallmatrix}\right)$ (which corresponds to (2.19)).

Problem 2.2 The saddle point problem with $A = \left(\begin{smallmatrix} \frac{1}{2}L^T L & 0 \\ 0 & \frac{1}{2}E \end{smallmatrix}\right)$ (which corresponds to (2.15)).

We use the stopping criterion

$$\big\|u - u^*\big\|_{L_2} = \left(h \sum_{i=1}^{n} \big(u_i - u_i^*\big)^2\right)^{1/2} < 10^{-4},$$

where $h = n^{-1}$ is the mesh step and u^* is the known exact solution, and the initial guess $\lambda = 0$. Table 2.1 demonstrates the dependence of the number of iterations n_{it} upon the iterative parameter and the number of the grid nodes for Problem 2.1.

For Problem 2.2 the optimal iterative parameter was found $\tau = 0.4$ and the number of iterations to achieve the accuracy $\|u - u^*\|_{L_2} < 10^{-4}$ for the grids with the number of nodes from $n = 50$ to $n = 500\,000$ was equal to 12.

Table 2.2 *Left*: The Uzawa method with the preconditioner B_λ equals to the unit matrix for Problem 2.3, the initial guess $\lambda = 0$. *Right*: Algorithm 2 for Problem 2.4, corresponding to the augmented Lagrangian method, the initial guess $\lambda = 0$, $p = 0$

n	200					400
τ	1.2	1.3	1.4	1.5	1.6	1.3
n_{it}	11	11	13	17	23	11

n	200	400	500
τ	1.3	1.3	1.3
n_{it}	9	9	9

Now we consider two-dimensional variational inequalities with linear differential operators

$$\int_\Omega \nabla u \cdot \nabla(v-u)\,\mathrm{d}x \geq \int_\Omega f(v-u)\,\mathrm{d}x, \quad \forall v \in K,$$

$$K = \left\{ u \in H_0^1(\Omega) : \left|\frac{\partial u}{\partial x_1}\right| \leq 1, \ \left|\frac{\partial u}{\partial x_2}\right| \leq 1 \text{ in } \Omega \right\}; \tag{2.30}$$

$$\int_\Omega \nabla u \cdot \nabla(v-u)\,\mathrm{d}x + \int_\Omega |\nabla v| - |\nabla u|\,\mathrm{d}x \geq \int_\Omega f(v-u)\,\mathrm{d}x \quad \forall v \in H_0^1(\Omega). \tag{2.31}$$

We set $\Omega = (0,1) \times (0,1)$ and construct finite difference approximations on uniform grids. These finite difference schemes can be written in the form of the inclusion $L^T L u + L^T \partial\theta(Lu) \ni f$, where the rectangular matrix L corresponds to the approximation of the gradient operator. We have studied the following two saddle point problems:

Problem 2.3 2D saddle point problem with the matrix $A = \left(\begin{smallmatrix} L^T L & 0 \\ -L & E \end{smallmatrix}\right)$.

Problem 2.4 2D saddle point problem with the matrix $A = \left(\begin{smallmatrix} 2L^T L & -L^T \\ -L & E \end{smallmatrix}\right)$ (which corresponds to the augmented Lagrangian method with $r = 1$).

We use the stopping criterion

$$\left\|u - u^*\right\|_{L_2} = \left(h^2 \sum_{i,j=1}^{n} \left(u_{ij} - u_{ij}^*\right)^2 \right)^{1/2} < 10^{-3},$$

where $n = h^{-1}$ is the number of nodes in one direction and u^* is the known exact solution. Table 2.2 contains results for the variational inequality (2.30).

For the discrete saddle point problems corresponding to (2.31) the results were similar. Namely, for both aforementioned methods and grids with the number of nodes $n = 100, 200, 400$ the accuracy $\|u - u^*\|_{L_2} < 10^{-3}$ was achieved within 19 iterations for $\tau = 1.2$, which was found as numerically optimal.

Table 2.3 2D non-linear saddle point problem; $C = 10$, $\tau = 1/2$, $n = 500$

n_{it}	1	10	20	30	40	50	60	70
$\|r_\lambda\|$	0.7137	0.1144	0.0248	0.0095	0.0050	0.0030	0.0020	0.0015
δu	0.0829	0.0123	0.0058	0.0014	0.0009	0.0005	0.0003	0.0001

Finally, we consider a two-dimensional variational inequality associated with the non-linear differential operator

$$\int_\Omega k\big(|\nabla u|\big)\nabla u \cdot \nabla(v-u)\,\mathrm{d}x \geq C\int_\Omega (v-u)\,\mathrm{d}x, \quad \forall v \in K, \tag{2.32}$$

where $\Omega = (0,1)\times(0,1)$, $k(t)t = \sqrt{t}$ and $K = \{u \in H_0^1(\Omega) : |\nabla u(x)| \leq 1 \text{ in } \Omega\}$. We constructed a finite difference approximation of (2.32) on the uniform grid. According to the theory the iterative parameter was taken $\tau = 1/2$. Since the exact solution was not known we estimated the norms of the residuals $\|r_\lambda\|_{L_2}$ (see the estimate (2.10)). Calculations were made for different amount of nodes in one direction. For all grids, we observed typical dependence of norms of the residuals upon the iteration number: very fast decreasing during the first iterations with further deceleration. After 20–25 iterations the norm $\|u^k - u^{k-1}\|_{L_2}$ became very close to zero and the vector u^k could be taken as the exact solution. The calculation results for the case $n = 500$ are given in Table 2.3, where $\delta u = \|u^k - u^{100}\|_{L_2}$ is the norm of the difference between the current iteration and the 100th iteration which was taken as the exact solution.

In the computations performed for 1D and 2D variational inequalities, the following features were observed:

- The dependence of the rate of convergence for the method (2.4) on the parameters r and $\tau = \tau(r)$ was quite low;
- The number of iterations did not depend on the mesh size $h = 1/n$;
- In all cases the Uzawa-type method (2.4) applied to transformed saddle point problems with the block triangle A was similar by a rate of convergence to Algorithm 2 applied to the saddle point problem constructed via the augmented Lagrangian technique.

2.4 Application to Optimal Control Problems

Consider the following elliptic boundary value problem:

$$\int_\Omega \sum_{i,j=1}^{2}\left(a_{ij}\frac{\partial y}{\partial x_j}\frac{\partial z}{\partial x_i} + a_0 yz\right)\mathrm{d}x = \int_\Omega (f+\chi_0 u)z\,\mathrm{d}x \quad \forall z \in H_0^1(\Omega). \tag{2.33}$$

Here $\Omega_0 \subseteq \Omega$, $\chi_0 \equiv \chi_{\Omega_0}$ is the characteristic function of the domain Ω_0, the function $f \in L_2(\Omega)$ is fixed, while $u \in L_2(\Omega_0)$ is a variable control function. Coefficients $a_{ij}(x)$ and $a_0(x)$ are continuous in $\overline{\Omega}$ and satisfy the following ellipticity assumptions:

$$\sum_{i,j=1}^{2} a_{ij}(x)\xi_j\xi_i \geq c_0 \sum_{i=1}^{2} \xi_i^2, \qquad a_0(x) \geq 0 \quad \forall x \in \overline{\Omega},\ c_0 = \text{const} > 0.$$

Define the goal functional

$$J(y, f) = \frac{1}{2}\int_{\Omega_1} (y - y_d)^2 \,\mathrm{d}x + \frac{1}{2}\int_{\Omega_0} u^2 \,\mathrm{d}x$$

with a given function $y_d(x) \in L_2(\Omega_1)$, $\Omega_1 \subseteq \Omega$, and the sets of the constraints

$$Y_{ad} = \big\{y \in V : y(x) \geq 0 \ \forall x \in \Omega\big\}, \qquad U_{ad} = \big\{u \in L_2(\Omega_0) : \big|u(x)\big| \leq u_d \ \forall x \in \Omega_0\big\}.$$

The optimal control problem reads as follows:

$$\min_{(y,u)\in Z} J(y, u), \qquad Z = \big\{(y, u) : y \in Y_{ad},\ u \in U_{ad},\ \text{Eq. (2.33) holds}\big\}. \tag{2.34}$$

We suppose that the set Z is non-empty. Then, the problem (2.34) has a unique solution (cf., e.g., [20]).

Construct a finite element approximation of the problem (2.34) in the case of polygonal domains Ω, Ω_0 and Ω_1. Let a triangulation of Ω be consistent with Ω_0 and Ω_1. Define the spaces of the continuous and piecewise linear functions (linear on each triangle of the triangulation) on the domain Ω ($V_h \subset H_0^1(\Omega)$) and on the subdomains Ω_0 and Ω_1. Let functions f, u and y_d be continuous and f_h, u_h and y_{dh} be their piecewise linear interpolations. We use the quadrature formulas

$$\int_e g(x)\,\mathrm{d}x \approx S_e(g) = \frac{1}{3}|e| \sum_{\alpha=1}^{3} g(x_\alpha),$$

$$S_\Omega(g) = \sum_{e\in T_h} S_e(g), \qquad S_{\Omega_i}(g) = \sum_{e\in T_h^i} S_e(g),$$

where x_α are the vertices of e, and $|e| = \operatorname{meas} e$. Finite element approximations of the state equation, the goal function, and the constraints are as follows:

$$S_\Omega\left(\sum_{i,j=1}^{2} a_{ij}\frac{\partial y_h}{\partial x_j}\frac{\partial z_h}{\partial x_i} + a_0 y_h z_h\right) = S_\Omega(f_h z_h) + S_{\Omega_0}(u_h z_h) \quad \forall z_h \in V_h, \tag{2.35}$$

$$J_h(y_h, u_h) = \frac{1}{2}S_{\Omega_1}\big((y_h - y_{dh})^2\big) + \frac{1}{2}S_{\Omega_0}\big(u_h^2\big),$$

$$Y_{ad}^h = \big\{y_h \in V_h :\ y_h(x) \geq 0 \ \text{in}\ \Omega\big\}, \qquad U_{ad}^h = \big\{u_h : \big|u_h(x)\big| \leq u_d \ \text{in}\ \overline{\Omega}_0\big\}.$$

The state equation (2.35) has a unique solution y_h and the following stability inequality holds:

$$S_{\Omega}^{1/2}\big(|y_h|^2\big) \le k_a\big(S_{\Omega}^{1/2}\big(f_h^2\big) + S_{\Omega_0}^{1/2}\big(u_h^2\big)\big) \tag{2.36}$$

with a constant k_a independent on h. The finite element approximation of the optimal control problem (2.34) is

$$\begin{cases} \min\limits_{(y_h,u_h)\in Z_h} J_h(y_h,u_h), \\ Z_h = \{(y_h,u_h):\ y_h \in Y_{ad}^h, u_h \in U_{ad}^h,\ \text{Eq. (2.35) holds}\}. \end{cases} \tag{2.37}$$

To obtain the matrix-vector form of (2.37), we define the vectors of nodal values $y \in \mathbb{R}^{N_y}$, $u \in \mathbb{R}^{N_u}$ and the matrices

$$L \in \mathbb{R}^{N_y\times N_y} : (Ly,z) = S_{\Omega}\left(\sum_{i,j=1}^{2} a_{ij}\frac{\partial y_h}{\partial x_j}\frac{\partial z_h}{\partial x_i} + a_0 y_h z_h\right),$$

$$S \in \mathbb{R}^{N_y\times N_u} : (Su,z) = S_{\Omega_0}(u_h z_h), \qquad K \in \mathbb{R}^{N_y\times N_y} : (Ky,z) = S_{\Omega_1}(y_h z_h),$$

$$M \in \mathbb{R}^{N_y\times N_y} : (Mf,z) = S_{\Omega}(f_h z_h), \qquad M_0 \in \mathbb{R}^{N_u\times N_u} : (M_0 u,v) = S_{\Omega_0}(u_h v_h).$$

Then, the discrete optimal control problem can be written in the form

$$\min_{Ly=Mf+Su}\left\{\frac{1}{2}(Ky,y) - (Ky_d,y) + \theta(y) + \frac{1}{2}(M_0u,u) + \varphi(u)\right\},$$

where $\theta(y) = I_{Y_{ad}}(y)$ and $\varphi(u) = I_{U_{ad}}(u)$ are the indicator functions of the sets $Y_{ad} = \{y \in \mathbb{R}^{N_y} : y_i \ge 0\ \forall i\}$ and $U_{ad} = \{u \in \mathbb{R}^{N_u} : |u_i| \le u_d\ \forall i\}$, respectively. The corresponding saddle point problem reads as follows:

$$\begin{pmatrix} K & 0 & -L^T \\ 0 & M_0 & S^T \\ -L & S & 0 \end{pmatrix}\begin{pmatrix} y \\ u \\ \lambda \end{pmatrix} + \begin{pmatrix} \partial\theta(y) \\ \partial\varphi(u) \\ 0 \end{pmatrix} \ni \begin{pmatrix} Ky_d \\ 0 \\ -Mf \end{pmatrix}. \tag{2.38}$$

In the problem (2.38), the stiffness matrix L is positive definite, and $M > 0$, $M_0 > 0$, $K \ge 0$ are diagonal matrices. The main feature of (2.38) is that K is a degenerate matrix. We transform the system (2.38) to obtain a positive definite and block triangle left upper 2×2 block. To this end we add to the first inclusion in (2.38) the last equation multiplying by $-rML^{-1}$, $r > 0$, and obtain the saddle point problem

$$\begin{pmatrix} A[r] & -C^T \\ -C & 0 \end{pmatrix}\begin{pmatrix} x \\ \lambda \end{pmatrix} + \begin{pmatrix} \partial\Theta(x) \\ 0 \end{pmatrix} \ni \begin{pmatrix} \tilde{g} \\ -Mf \end{pmatrix} \tag{2.39}$$

with

$$A[r] = \begin{pmatrix} K + rM & -rML^{-1}S \\ 0 & M_0 \end{pmatrix}, \qquad \partial\Theta(x) = \begin{pmatrix} \partial\theta(y) \\ \partial\varphi(u) \end{pmatrix}$$

and $\tilde{g} = (\tilde{f},0)^T$, $\tilde{f} = Ky_d + rML^{-1}Mf$.

Lemma 2.7 *Let* $0 < r < \frac{4}{k_a^2}$, *where the constant* k_a *is defined in* (2.36). *Then, the matrix* $A[r]$ *is an energy equivalent to* $A^0 = \left(\begin{smallmatrix} M & 0 \\ 0 & M_0 \end{smallmatrix}\right)$ *with constants depending only on* r. *In particular,*

$$\big(A[r]x, x\big) \geq \alpha\big(A^0 x, x\big), \quad \alpha = \alpha(r, k_a) > 0.$$

We solve (2.39) by using the iterative Uzawa-type method (2.4) with the preconditioner $B_\lambda = L\,M^{-1}\,L^T$:

$$\begin{aligned} (K + rM)y^{k+1} + \partial\theta\big(y^{k+1}\big) - rML^{-1}Su^{k+1} &\ni L^T\lambda^k + \tilde{f}, \\ M_0 u^{k+1} + \partial\varphi\big(u^{k+1}\big) &\ni -S^T\lambda^k, \\ \frac{1}{\tau}LM^{-1}L^T\big(\lambda^{k+1} - \lambda^k\big) + Ly^{k+1} - Su^{k+1} &\ni Mf. \end{aligned} \tag{2.40}$$

Theorem 2.6 ([18]) *The iterative method* (2.40) *converges if*

$$0 < \tau < \frac{2\alpha}{k_a^2 + 1}.$$

Along with the iterative method (2.40) we can use the gradient method for the regularized problem. Namely, let us change the indicator function $\theta(y) = I_{Y_{ad}}(y)$ of the constraint set $Y_{ad} = \{y \in \mathbb{R}^{N_y} : y_i \geq 0\ \forall i\}$ by the differentiable function

$$\theta_\varepsilon(y) = \frac{1}{\varepsilon}\big(My^-, y^-\big).$$

For the corresponding regularized saddle point problem we can apply the "traditional" gradient method

$$\begin{aligned} Ly^{k+1} &= Su^k + Mf, \\ L^T\lambda^{k+1} &= (K + rM)y^{k+1} + \nabla\theta_\varepsilon\big(y^{k+1}\big) - rML^{-1}Su^k - \tilde{f}, \\ M_0\frac{u^{k+1} - u^k}{\tau} &+ M_0 u^{k+1} + \partial\varphi\big(u^{k+1}\big) + S^T\lambda^{k+1} \ni 0. \end{aligned} \tag{2.41}$$

Theorem 2.7 ([19]) *The iterative method* (2.41) *converges if*

$$0 < \tau < \frac{2\varepsilon}{k_a^2(1+\varepsilon) + r\varepsilon}.$$

When implementing any of the iterative methods (2.40) or (2.41) we have to solve the systems of linear equations with matrices L and L^T, and to solve two inclusions with diagonal operators $M_0 + \partial\varphi$ and $K + rM + \partial\theta$.

Table 2.4 The Uzawa-type method for Problem 2.5, $y = 3(\sin(6\pi x_1 x_2))^+$

n_{it}	$n = 100$, $F^* = 1.70$		$n = 300$, $F^* = 1.68$		$n = 500$, $F^* = 1.68$	
	F	Err	F	Err	F	Err
1	0	0.05	0	0.048604	0	0.048052
2	1.71	0.0001	1.68	0.00012238	1.68	0.00012111
3	1.70	3×10^{-7}	1.68	3.1×10^{-7}	1.68	3.1×10^{-7}
4	1.70	1.69×10^{-7}	1.68	6.79×10^{-8}	1.68	1.47×10^{-7}
5	1.70	1.69×10^{-7}	1.68	6.79×10^{-8}	1.68	1.47×10^{-7}

2.4.1 Numerical Experiments

Problem 2.5 A control- and state-constrained optimal control problem with observation in the whole domain $\Omega = (0, 1) \times (0, 1)$: minimize the goal functional

$$\frac{1}{2}\int_\Omega y^2(x)\,\mathrm{d}x + \frac{1}{2}\int_\Omega u^2(x)\,\mathrm{d}x$$

under the constraints

$$\begin{aligned} -\Delta y = f + u, \quad x \in \Omega, \qquad & y(x) = 0, \quad x \in \partial\Omega, \\ y(x) \geq 0, \qquad x \in \Omega, \qquad & \left|u(x)\right| \leq 1, \quad x \in \Omega. \end{aligned} \tag{2.42}$$

We constructed a finite difference approximation of this problem on the uniform grid. The corresponding saddle point problem has the form (2.38) with unit matrices K, M_0 and S. Therefore, we can use the preconditioned Uzawa-type method (2.40) for solving this saddle point problem without its transformation. The results of the calculations are reported in Table 2.4, where $F^* = J(y, u)$ is the value of the discrete goal function on the known exact solution (y, u) $(y = 3(\sin(6\pi x_1 x_2))^+$ for the corresponding grid), while $F = J(y^k, v^k)$ is its value on the current iteration; $Err = (\|y^k - y\|_{L_2}^2 + \|u^k - u\|_{L_2}^2)^{\frac{1}{2}}$.

Problem 2.6 A control- and state-constrained optimal control problem with observation in the part $\Omega_1 = (0, 0.7) \times (0, 1)$ of the domain $\Omega = (0, 1) \times (0, 1)$: minimize the goal functional

$$\frac{1}{2}\int_{\Omega_1} y^2(x)\,\mathrm{d}x + \frac{1}{2}\int_\Omega u^2(x)\,\mathrm{d}x$$

under the constraints (2.42). We constructed a finite difference approximation of this problem on the uniform grid. The corresponding saddle point problem has the form (2.38) with the degenerate matrix K. We transformed it to the problem of the form (2.39) with $r = 1$ and applied the Uzawa-type method (2.40) for its solution. The corresponding calculation results are included in Table 2.5.

Table 2.5 The Uzawa-type method for Problem 2.6

n_{it}	$n = 100$, $F^* = 2.7783$		$n = 200$, $F^* = 2.7897$		$n = 500$, $F^* = 2.7965$	
	F	Err	F	Err	F	Err
1	0.6836	0.8652	0.6900	0.8705	0.6974	0.8736
2	1.5378	0.4285	1.5574	0.4311	1.5689	0.4327
3	2.0928	0.2113	2.1194	0.2125	2.1352	0.2133
4	2.4020	0.1073	2.4326	0.1080	2.4507	0.1084
5	2.5645	0.0580	2.5972	0.0583	2.6165	0.0585
6	2.6477	0.0336	2.6814	0.0337	2.7013	0.0338
7	2.6897	0.0214	2.7240	0.0215	2.7442	0.0215
8	2.7109	0.0153	2.7454	0.0153	2.7658	0.0154
9	2.7215	0.0122	2.7561	0.0123	2.7766	0.0123
10	2.7268	0.0107	2.7615	0.0107	2.7820	0.0107
⋮	⋮	⋮	⋮	⋮	⋮	⋮
50	2.7320	0.0088	2.7669	0.0088	2.7874	0.0088

Problem 2.7 A state-constrained optimal control problem with observation in the whole domain: minimize the goal functional

$$J(y,u) = \frac{1}{2}\int_{\Omega}(y - y_d)^2\,\mathrm{d}x + \frac{1}{2}\int_{\Omega} u^2\,\mathrm{d}x$$

under the constraints

$$-\Delta y = f + u, \quad x \in \Omega, \qquad y(x) = 0, \quad x \in \partial\Omega,$$
$$y(x) \le 0.5, \quad x \in \Omega.$$

We constructed a finite difference approximation on the uniform grid and applied the Uzawa-type method (2.40) and the gradient method (2.41) for solving the corresponding discrete saddle point problems. We compared the calculated iterations with the exact solution y, calculated by using a great deal of convergent iterations. Table 2.6 contains the results for the case $f = 20$, $h = 10^{-2}$, $F^* = 44.1789$. The notations are $Err_y = \|y - y^k\|$, $\delta y^k = \|y^{k-1} - y^k\|$.

Along with the Uzawa-type and regularization methods, we have also applied the Douglas-Rachford splitting method for solving state-constrained optimal control problems. We have found that none of the methods could be defined as the efficient one in all situations. More numerical experiments should be made to define the classes of the optimal control problems and the corresponding iterative methods which are the most efficient for their solving.

Table 2.6 Uzawa-type and gradient methods for Problem 2.7

n_{it}	Uzawa method with $\tau = 1.8$			Gradient method with $\varepsilon = 10^{-5}$, $\tau = 2 \times 10^{-5}$		
	F	Err_y	δy^k	F	Err_y	δy^k
1	0	0.3629	0.0750	0.3396	21.7302	0.8665
2	0.1089	0.1552	0.4683	0.3406	21.5275	0.0455
3	0.0984	0.1085	0.1350	0.3435	21.3268	0.0452
4	0.1092	0.1229	0.1110	0.3482	21.1279	0.0450
5	0.1090	0.0986	0.0953	0.3547	20.9310	0.0448
6	0.1215	0.1125	0.0827	0.3630	20.7361	0.0446
7	0.1267	0.0971	0.0738	0.3731	20.5430	0.0443
8	0.1405	0.1069	0.0670	0.3848	20.3517	0.0441
9	0.1499	0.0970	0.0624	0.3983	20.1623	0.0439
10	0.1654	0.1034	0.0590	0.4134	19.9748	0.0437
⋮	⋮	⋮	⋮	⋮	⋮	⋮
300	21.1177	0.0568	0.0157	23.3858	1.5660	0.0108
⋮	⋮	⋮	⋮	⋮	⋮	⋮
500	31.2662	0.0425	0.0064	33.4540	0.3330	0.0045

References

1. Benzi M, Golub G, Liesen J (2005) Numerical solution of saddle point problems. Acta Numer 14:1–137
2. Bergounioux M (1993) Augmented Lagrangian method for distributed optimal control problems with state constraints. J Optim Theory Appl 78(3):493–521
3. Bergounioux M, Haddou V, Hintermuller M, Kunisch K (2000) A comparison of a Moreau-Yosida-based active set strategy and interior point methods for constrained optimal control problems. SIAM J Optim 11(2):495–521
4. Bergounioux M, Kunisch K (2002) Primal-dual strategy for state-constrained optimal control problems. Comput Optim Appl 22(2):193–224
5. Biegler LT, Ghattas O, Heinkenschloss M, van Bloemen Waanders B (eds) (2003) Large-scale PDE-constrained optimization, Santa Fe, NM, 2001. Lect Notes Comput Sci Eng, vol 30. Springer, Berlin
6. Bychenkov Yu, Chizhonkov E (2010) Iterative solution methods for saddle point problems. Binom, Moscow. In Russian
7. Ciarlet PG, Lions J-L (eds) (1991) Handbook of numerical analysis. Finite element methods, vol II. North-Holland, Amsterdam
8. Fortin M, Glowinski R (1983) Augmented Lagrangian methods. North-Holland, Amsterdam
9. Glowinski R, Le Tallec P (1989) In: Augmented Lagrangian and operator-splitting methods in nonlinear mechanics. SIAM studies in applied mathematics, vol 9. SIAM, Philadelphia
10. Glowinski R, Lions J-L, Trémolières R (1976) Analyse numérique des inéquations variationnelles. Dunod, Paris
11. Graser C, Kornhuber R (2009) Newton methods for set-valued saddle point problems. SIAM J Numer Anal 47(2):1251–1273
12. Herzog R, Sachs E (2010) Preconditioned conjugate gradient method for optimal control problems with control and state constraints. SIAM J Matrix Anal Appl 31(5):2291–2317

13. Hintermüller M, Hinze M (2009) Moreau-Yosida regularization in state constrained elliptic control problems: error estimates and parameter adjustment. SIAM J Numer Anal 47(3):1666–1683
14. Hinze M, Schiela A (2011) Discretization of interior point methods for state constrained elliptic optimal control problems: optimal error estimates and parameter adjustment. Comput Optim Appl 48(3):581–600
15. Ito K, Kunisch K (2003) Semi-smooth Newton methods for state-constrained optimal control problems. Syst Control Lett 50(3):221–228
16. Laitinen E, Lapin A, Lapin S (2010) On the iterative solution for finite-dimensional inclusions with applications to optimal control problems. Comput Methods Appl Math 10(3):283–301
17. Laitinen E, Lapin A, Lapin S (2011) Iterative solution methods for variational inequalities with nonlinear main operator and constraints to gradient of solution. Preprint, University of Oulu
18. Lapin A (2010) Preconditioned Uzawa-type methods for finite-dimensional constrained saddle point problems. Lobachevskii J Math 31(4):309–322
19. Lapin AV, Khasanov MG (2010) The solution of a state constrained optimal control problem by the right-hand side of an elliptic equation. Kazan Gos Univ Uchen Zap Ser Fiz-Mat Nauki 152(4):56–67. http://mi.mathnet.ru/uzku884. In Russian
20. Lions J-L (1971) Optimal control of systems governed by partial differential equations. Springer, New York
21. Prüfert U, Tröltzsch F, Weiser M (2008) The convergence of an interior point method for an elliptic control problem with mixed control-state constraints. Comput Optim Appl 39(2):183–218

Chapter 3
Analytical-Numerical Methods for Hidden Attractors' Localization: The 16th Hilbert Problem, Aizerman and Kalman Conjectures, and Chua Circuits

Gennady A. Leonov and Nikolay V. Kuznetsov

Abstract This survey is devoted to analytical-numerical methods for hidden attractors' localization and their application to well-known problems and systems. From the computation point of view, in nonlinear dynamical systems the attractors can be regarded as *self-exciting* and *hidden attractors*. Self-exciting attractors can be localized numerically by the following *standard computational procedure*: after a transient process a trajectory, started from a point of an unstable manifold in a small neighborhood of unstable equilibrium, reaches an attractor and computes it. In contrast, a hidden attractor is an attractor whose basin of attraction does not contain neighborhoods of equilibria. In well-known Van der Pol, Belousov-Zhabotinsky, Lorenz, Chua, and many other dynamical systems classical attractors are self-exciting attractors and can be obtained numerically by the standard computational procedure. However, for localization of hidden attractors it is necessary to develop special analytical-numerical methods, in which at the first step the initial data are chosen in a basin of attraction and then the numerical localization (visualization) of the attractor is performed. The simplest examples of hidden attractors are internal nested limit cycles (hidden oscillations) in two-dimensional systems (see, e.g., the results concerning the second part of the *16th Hilbert's problem*). Other examples of hidden oscillations are counterexamples to *Aizerman's conjecture* and *Kalman's conjecture* on absolute stability in the automatic control theory (a unique stable equilibrium coexists with a stable periodic solution in these counterexamples). In 2010, for the first time, a *chaotic hidden attractor* was computed first by the authors in a *generalized Chua's circuit* and then one chaotic hidden attractor was discovered in a *classical Chua's circuit*.

G.A. Leonov · N.V. Kuznetsov
St. Petersburg State University, Universitetsky pr. 28, St. Petersburg 198594, Russia

G.A. Leonov
e-mail: leonov@math.spbu.ru

N.V. Kuznetsov (✉)
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland
e-mail: nkuznetsov239@gmail.com

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_3,

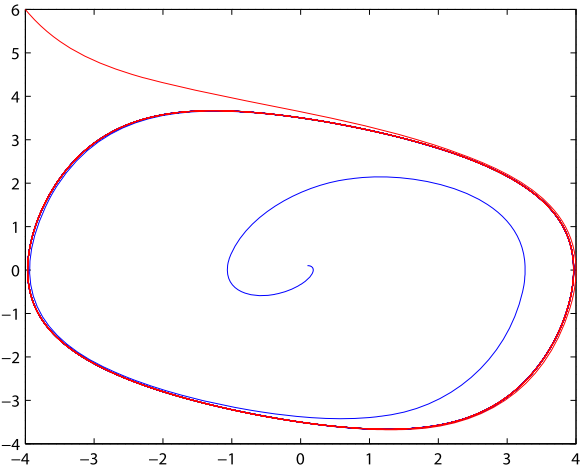


Fig. 3.1 Numerical localization of the limit cycle in the Rayleigh system

3.1 Introduction

In the first half of last century, during the initial period of development of the theory of nonlinear oscillations [2, 11, 32, 33], main attention has been given to analysis and synthesis of oscillating systems, for which the existence problem of oscillations can be solved relatively easily. The structure of many mechanical, electromechanical, and electronic systems is such that the existence of oscillations in them is almost obvious, namely the oscillations are excited from unstable equilibria. From the computational point of view it means that one can use a *standard numerical method*, in which after a transient process a trajectory, started from a point of an unstable manifold in a small neighborhood of equilibrium, reaches an attractor and identifies it.

Consider the following classical examples.

Example 3.1 (The Rayleigh string oscillator) In studying string oscillations [31] Rayleigh discovered first that in the two-dimensional nonlinear dynamical system

$$\ddot{x} - \left(a - b\dot{x}^2\right)\dot{x} + x = 0, \tag{3.1}$$

undamped vibrations (namely limit cycles—this term was introduced later by Poincare) can arise. A well-known generalization of this system is the Van der Pol equation [34] that describes the nonlinear electrical circuits used in radio engineering. The result of the simulation of this system (3.1) for $a = 1$, $b = 0.1$ is presented in Fig. 3.1.

Example 3.2 (The Belousov-Zhabotinsky (BZ) reaction) In 1951 B.P. Belousov discovered the first oscillations in the chemical reactions [3]. Consider one of the

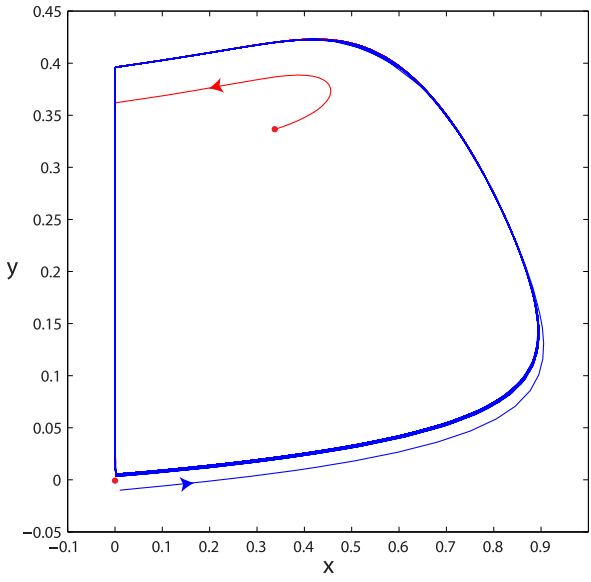


Fig. 3.2 Numerical localization of the limit cycle in the Belousov-Zhabotinsky model

Belousov-Zhabotinsky dynamical models

$$\begin{aligned} \varepsilon \dot{x} &= x(1-x) + \frac{f(q-x)}{q+x} z, \\ \dot{z} &= x - z, \end{aligned} \tag{3.2}$$

and perform its simulation, using standard parameters: $f = 2/3$, $q = 8 \times 10^{-4}$, $\varepsilon = 4 \times 10^{-2}$ (see Fig. 3.2).

Consider now the examples of numerical localization of well-known chaotic attractors in three-dimensional dynamical models.

Example 3.3 (The Lorenz system) Consider the Lorenz system [27]

$$\begin{aligned} \dot{x} &= \sigma(y - x), \\ \dot{y} &= x(\rho - z) - y, \\ \dot{z} &= xy - \beta z, \end{aligned} \tag{3.3}$$

and carry out its simulation with standard parameters $\sigma = 10$, $\beta = 8/3$, $\rho = 28$ (see Fig. 3.3). Here the computed trajectory is started from a small neighborhood of an unstable zero stationary point.

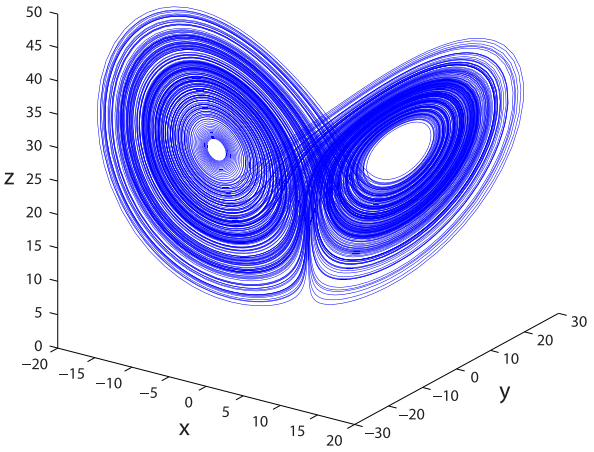


Fig. 3.3 Numerical localization of a chaotic attractor in the Lorenz system

Example 3.4 (The Chua system) Consider the classical Chua circuit [7] and its dynamical model in dimensionless coordinates

$$\begin{aligned} \dot{x} &= \alpha(y - x) - \alpha f(x), \\ \dot{y} &= x - y + z, \\ \dot{z} &= -(\beta y + \gamma z). \end{aligned} \tag{3.4}$$

Here the function

$$f(x) = m_1 x + (m_0 - m_1)\mathrm{sat}(x) \tag{3.5}$$

characterizes a nonlinear element called the Chua diode. In this system, strange attractors [29] then called the Chua attractors were discovered. To date all the known classical Chua attractors are those excited from unstable equilibria. This makes it possible to compute different Chua attractors with relative ease [5]. Perform the simulation of the Chua attractor with the following parameters: $\alpha = 9.35$, $\beta = 14.79$, $\gamma = 0.016$, $m_0 = -1.1384$, $m_1 = 0.7225$ (see Fig. 3.4).

Here, in all examples, the limit cycles and attractors are those excited from unstable equilibria (i.e., self-excited attractors).

3.2 Hidden Oscillations and Hidden Attractors

In the middle of the last century, oscillations of another type were found, so-called *hidden oscillations*: the oscillations, the existence of which is not obvious. They are not "connected" with equilibrium (i.e. in this case it is impossible to localize a periodic solution by the computing of trajectory with the initial data from a small

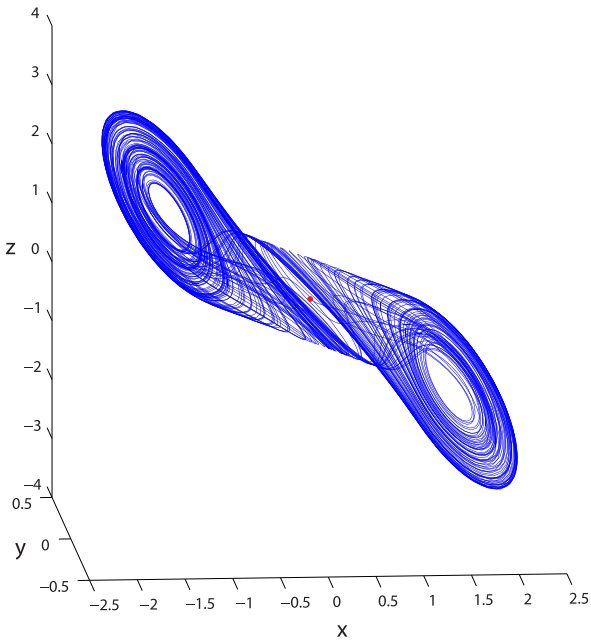


Fig. 3.4 Numerical localization of a chaotic attractor in the Chua circuit

neighborhood of equilibrium). In addition, in this case it is unlikely that the integration of trajectories with random initial data will lead to localization of such hidden oscillation since the basin of attraction can be very small and the considered system dimension can be large.

For the first time the problem of finding hidden oscillations arose in the 16th Hilbert problem (1900) for two-dimensional polynomial systems. For more than a century, in the framework of the solution of this problem, the numerous theoretical and numerical results were obtained. However, the problem is still far from being resolved even for the simple class of quadratic systems. In the 1940s and 1950s, academician A.N. Kolmogorov became the initiator of a few hundred of the following computational experiments [16]: he asked students (at Moscow State University) to find limit cycles in two-dimensional quadratic systems with randomly chosen coefficients. The result was absolutely unexpected: limit cycles were not found in any of the experiments, though it is known that quadratic systems with limit cycles form open domains in the space of coefficients and, therefore, for a random choice of polynomial coefficients, the probability of hitting in these sets is positive.

Note that numerical localization of small and nested limit cycles [13, 16, 20, 22, 24, 25] is a difficult problem.

Example 3.5 (Four limit cycles in a quadratic system) Nowadays the application of special analytical-numerical methods [17, 25] allows one to visualize four limit cycles in a quadratic system [12].

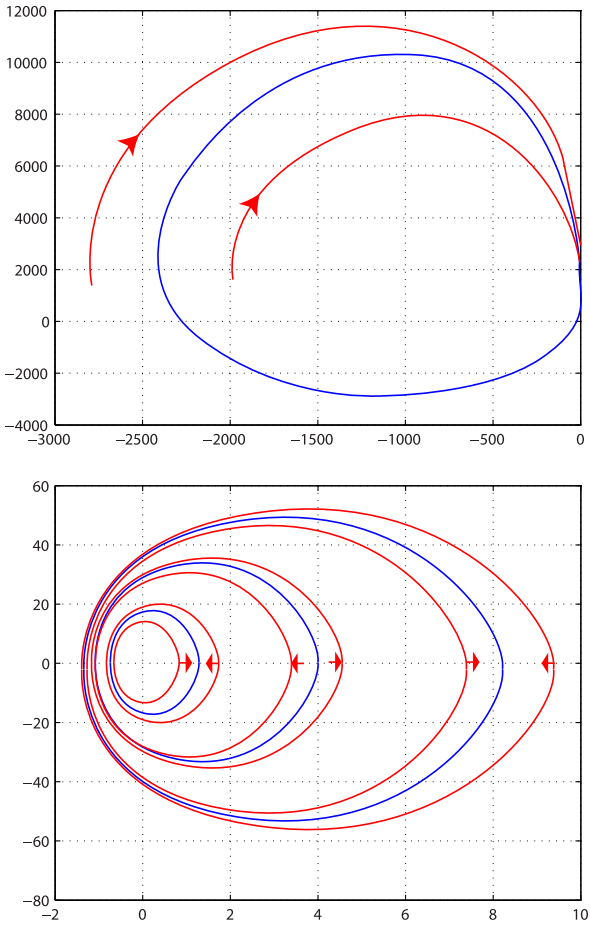


Fig. 3.5 Visualization of four limit cycles in a polynomial quadratic system

Consider the following quadratic system:

$$\begin{aligned} \frac{dx}{dt} &= x^2 + xy + y, \\ \frac{dy}{dt} &= a_2x^2 + b_2xy + c_2y^2 + \alpha_2x + \beta_2y. \end{aligned} \tag{3.6}$$

In Fig. 3.5 for the coefficients

$$b_2 = 2.7, \qquad c_2 = 0.4, \qquad a_2 = -10, \qquad \alpha_2 = -437.5, \qquad \beta_2 = 0.003,$$

three "large" (normal size) limit cycles around the zero point and one "large" limit cycle to the left of the straight line $x = -1$ are visualized.

Further the problem of analysis of hidden oscillations arose in engineering problems of automatic control. In 1961 Gubar' [8] showed analytically the possibility of

existence of hidden oscillation in a two-dimensional system of a phase locked-loop with piecewise-constant impulse nonlinearity. In the 1950s and 1960s, the investigations of widely known Markus-Yamabe [28], Aizerman [1], and Kalman [9] conjectures on absolute stability had led to the finding of hidden oscillations in automatic control systems with a unique stable stationary point and the nonlinearity belonging to the sector of linear stability (see, e.g., [4, 6, 18, 30]).

Later, in 2010, for the first time, a *chaotic hidden attractor* was computed, by the authors, in a generalized Chua circuit [14] and then one chaotic hidden attractor was discovered in the classical Chua circuit [23].

Since the key factor, providing the possibility of computing the oscillation, is a basin of attraction, the following definition can be formulated.

Definition 3.1 Hidden attractors are those attractors whose basin of attraction does not contain neighborhoods of equilibria.

Here it is of the essence to consider a basin of attraction in forward and backward time since the computation in backward time may allow one to localize an unstable oscillation.

3.2.1 Analytical-Numerical Method for Localization of Hidden Oscillations in Multidimensional Dynamical Systems

For numerical localization of hidden oscillations the methods based on *homotopy* turned out to be the most effective ones. In this case a sequence of similar systems is considered such that for the first starting system the initial data for numerical localization of a periodic solution (starting periodic solution) can be obtained analytically and then the transformation of this starting periodic solution in the transition from one system to another is followed numerically.

Further we consider an effective analytical-numerical approach for localization of hidden oscillations in multidimensional dynamical systems, which are based on the method of a small parameter, the method of harmonic linearization (the describing function method), numerical methods, and an applied bifurcation theory.

Consider a system with one scalar[1] nonlinearity:

$$\frac{d\mathbf{x}}{dt} = \mathbf{P}\mathbf{x} + \mathbf{q}\psi\left(\mathbf{r}^*\mathbf{x}\right), \quad \mathbf{x} \in \mathbb{R}^n. \tag{3.7}$$

Here $\mathbf{P}$ is a constant $(n \times n)$-matrix, $\mathbf{q}, \mathbf{r}$ are constant n-dimensional vectors, * is a transposition operation, $\psi(\sigma)$ is a scalar function, and $\psi(0) = 0$. Define a coefficient of harmonic linearization k in such a way that the matrix

$$\mathbf{P}_0 = \mathbf{P} + k\mathbf{q}\mathbf{r}^* \tag{3.8}$$

[1] Vector nonlinearity can be considered similarly [26].

has a pair of purely imaginary eigenvalues $\pm i\omega_0$ ($\omega_0 > 0$) and the rest of its eigenvalues have negative real parts. Assume that such k exists. Rewrite the system (3.7) as

$$\frac{d\mathbf{x}}{dt} = \mathbf{P_0}\mathbf{x} + \mathbf{q}\varphi\big(\mathbf{r}^*\mathbf{x}\big), \tag{3.9}$$

where $\varphi(\sigma) = \psi(\sigma) - k\sigma$.

Introduce a finite sequence of functions $\varphi^0(\sigma), \varphi^1(\sigma), \ldots, \varphi^m(\sigma)$ such that the graphs of the neighboring functions $\varphi^j(\sigma)$ and $\varphi^{j+1}(\sigma)$ slightly differ from one another, the function $\varphi^0(\sigma)$ is small, and $\varphi^m(\sigma) = \varphi(\sigma)$. Using a smallness of the function $\varphi^0(\sigma)$, one can apply and mathematically strictly justify [15, 16, 18, 26] the method of harmonic linearization (the describing function method) for the system

$$\frac{d\mathbf{x}}{dt} = \mathbf{P_0}\mathbf{x} + \mathbf{q}\varphi^0\big(\mathbf{r}^*\mathbf{x}\big) \tag{3.10}$$

and find a stable nontrivial periodic solution $\mathbf{x}^0(t)$. For the localization of the oscillating solution (attractor) of the original system (3.9), we shall follow numerically the transformation of this periodic solution (a starting *oscillating attractor*, i.e. an attractor *not including equilibria*, denoted further by $\mathscr{A}_0$), with increasing j in passing from nonlinearity $\varphi^j(\sigma)$ to $\varphi^{j+1}(\sigma)$. Here two cases are possible:

Case 1: All the points of $\mathscr{A}_0$ are in the attraction domain of the attractor $\mathscr{A}_1$, being an oscillating attractor of the system

$$\frac{d\mathbf{x}}{dt} = \mathbf{P_0}\mathbf{x} + \mathbf{q}\varphi^j\big(\mathbf{r}^*\mathbf{x}\big) \tag{3.11}$$

with $j = 1$.

Case 2: In the change from the system (3.10) to the system (3.11) with $j = 1$ a loss of stability (bifurcation) and the vanishing of $\mathscr{A}_0$ are observed.

In Case 1 the solution $\mathbf{x}^1(t)$ can be determined numerically by starting a trajectory of the system (3.11) with $j = 1$ from the initial point $\mathbf{x}^0(0)$. If in the process of computation the solution $\mathbf{x}^1(t)$ has not fallen to an equilibrium and it is not increased indefinitely (here a sufficiently large computational interval $[0, T]$ should always be considered), then this solution reaches an attractor $\mathscr{A}_1$. Then it is possible to proceed to the system (3.11) with $j = 2$ and to perform a similar procedure of computation of $\mathscr{A}_2$, by starting a trajectory of the system (3.11) with $j = 2$ from the initial point $\mathbf{x}^1(T)$ and computing the trajectory $\mathbf{x}^2(t)$.

Proceeding this procedure and sequentially increasing j and computing $\mathbf{x}^j(t)$ (being a trajectory of the system (3.11) with the initial data $\mathbf{x}^{j-1}(T)$), one either arrives at the computation of $\mathscr{A}_m$ (being an attractor of the system (3.11) with $j = m$, i.e. the original system (3.9)), either, at a certain step, observes a loss of stability (bifurcation) and the vanishing of the attractor.

3.2.1.1 System Reduction

To determine the initial data $\mathbf{x}^0(0)$ of the starting periodic solution, one transforms the system (3.10) with nonlinearity $\varphi^0(\sigma)$ by the linear nonsingular transformation $\mathbf{S}$ to the form

$$\begin{aligned}\dot{y}_1 &= -\omega_0 y_2 + b_1\varphi^0\big(y_1 + \mathbf{c}_3^*\mathbf{y}_3\big),\\ \dot{y}_2 &= \omega_0 y_1 + b_2\varphi^0\big(y_1 + \mathbf{c}_3^*\mathbf{y}_3\big),\\ \dot{\mathbf{y}}_3 &= \mathbf{A}_3\mathbf{y}_3 + \mathbf{b}_3\varphi^0\big(y_1 + \mathbf{c}_3^*\mathbf{y}_3\big).\end{aligned} \tag{3.12}$$

Here y_1, y_2 are scalar values, $\mathbf{y}_3$ is an $(n-2)$-dimensional vector, $\mathbf{b}_3$ and $\mathbf{c_3}$ are $(n-2)$-dimensional vectors, b_1 and b_2 are real numbers, and $\mathbf{A}_3$ is an $((n-2)\times(n-2))$-matrix, all eigenvalues of which have negative real parts. Without loss of generality, it can be assumed that for the matrix $\mathbf{A}_3$ there exists a positive number $d > 0$ such that

$$\mathbf{y}_3^*\big(\mathbf{A}_3 + \mathbf{A}_3^*\big)\mathbf{y}_3 \le -2d|\mathbf{y}_3|^2, \quad \forall \mathbf{y}_3 \in \mathbb{R}^{n-2}. \tag{3.13}$$

In practice, for determining k and ω_0 the transfer function $W(p)$ of the system (3.7) is used:

$$W(p) = \mathbf{r}^*(\mathbf{P} - p\mathbf{I})^{-1}\mathbf{q},$$

where p is a complex variable. The number ω_0 is obtained from the equation $\operatorname{Im} W(i\omega_0) = 0$ and k is computed then by the formula $k = -(\operatorname{Re} W(i\omega_0))^{-1}$.

Let us write a transfer function of the system (3.10):

$$\mathbf{r}^*(\mathbf{P_0} - p\mathbf{I})^{-1}\mathbf{q} = \frac{\eta p + \theta}{p^2 + \omega_0^2} + \frac{R(p)}{Q(p)}, \tag{3.14}$$

and a transfer function of the system (3.12):

$$\frac{-b_1 p + b_2\omega_0}{p^2 + \omega_0^2} + \mathbf{c_3^*}(\mathbf{A}_3 - p\mathbf{I})^{-1}\mathbf{b}_3. \tag{3.15}$$

Here $\mathbf{I}$ is a unit matrix, η and θ are certain real numbers, $Q(p)$ is a stable polynomial of the degree $(n-2)$, $R(p)$ is a polynomial of a degree smaller than $(n-2)$. Suppose that the polynomials $R(p)$ and $Q(p)$ have no common roots. Since the systems (3.10) and (3.12) are equivalent, the transfer functions of these systems coincide. This implies the following relations:

$$\begin{aligned}&\eta = -b_1, \qquad \theta = b_2\omega_0,\\ &\mathbf{c_3^*}\mathbf{b}_3 + b_1 = \mathbf{r}^*\mathbf{q}, \qquad \frac{R(p)}{Q(p)} = \mathbf{c_3^*}(\mathbf{A}_3 - p\mathbf{I})^{-1}\mathbf{b}_3.\end{aligned} \tag{3.16}$$

3.2.1.2 Justification of Harmonic Balance in Non-critical Case

Consider the system (3.10) with differentiable[2] nonlinearity $\varphi^0(\sigma) = \varepsilon\varphi(\sigma)$, where ε is a small positive parameter.

Introduce the describing function

$$\Phi(a) = \int_0^{2\pi/\omega_0} \varphi\big(\cos(\omega_0 t)a\big)\cos(\omega_0 t)\,dt.$$

Theorem 3.1 ([6, 16]) *Let the number $a_0 > 0$ exist such that the conditions*

$$\Phi(a_0) = 0, \qquad b_1 \frac{d\Phi(a)}{da}\bigg|_{a=a_0} < 0 \tag{3.17}$$

are satisfied. Then for sufficiently small $\varepsilon > 0$ the system (3.12) with nonlinearity $\varphi^0(\sigma) = \varepsilon\varphi(\sigma)$ has a periodic solution of the form

$$\begin{aligned} y_1(t) &= \cos(\omega_0 t)y_1(0) + O(\varepsilon), \\ y_2(t) &= \sin(\omega_0 t)y_1(0) + O(\varepsilon), \qquad t \in [0, T] \\ \mathbf{y}_3(t) &= \exp(\mathbf{A}_3 t)\mathbf{y}_3(0) + \mathbf{O_{n-2}}(\varepsilon), \end{aligned} \tag{3.18}$$

with the initial data

$$y_1(0) = a_0 + O(\varepsilon), \qquad y_2(0) = 0, \qquad \mathbf{y}_3(0) = \mathbf{O_{n-2}}(\varepsilon) \tag{3.19}$$

and with the period

$$T = \frac{2\pi}{\omega_0} + O(\varepsilon).$$

Here $\mathbf{O_{n-2}}(\varepsilon)$ is an $(n-2)$-dimensional vector such that its components are $O(\varepsilon)$.

Taking into account the relations (3.16), this theorem can be reformulated in the following way.

Corollary 3.1 *Let the number $a_0 > 0$ exist such that the conditions*

$$\Phi(a_0) = 0, \qquad \eta \frac{d\Phi(a)}{da}\bigg|_{a=a_0} > 0 \tag{3.20}$$

are satisfied. Then for sufficiently small $\varepsilon > 0$ the system (3.10) with the transfer function (3.14) and the nonlinearity $\varphi^0(\sigma) = \varepsilon\varphi(\sigma)$ has a T-periodic solution such that

$$\mathbf{r}^*\mathbf{x}(t) = a_0\cos(\omega_0 t) + O(\varepsilon), \qquad T = \frac{2\pi}{\omega_0} + O(\varepsilon). \tag{3.21}$$

[2] There is similar consideration for piecewise- continuous function being Lipschitz on closed continuity intervals [16].

Theorem 3.1 coincides with the procedure of the search of stable periodic solutions by means of the standard describing function method (see, for example, [10]). Similar assertions can be proved in the case of vector nonlinearity [26].

It should be noted that the condition (3.20) cannot be satisfied in the case when conditions of the Aizerman and Kalman conjectures are fulfilled (i.e. nonlinearity φ belongs to the sector of linear stability). In this case the methods of harmonic balance and describing function lead to a wrong result, namely nonexistence of periodic solutions and global stability of unique equilibrium, but nowadays the counterexamples are well known [6, 16].

3.2.1.3 Justification of Harmonic Balance in the Critical Case

In 1957 R.E. Kalman formulated the following conjecture [9]:

Conjecture 3.1 Suppose that for all $k \in (\mu_1, \mu_2)$ a zero solution of the system (3.9) with $\varphi(\sigma) = k\sigma$ is asymptotically stable in the large (i.e., a zero solution is Lyapunov stable and any solution of the system (3.9) tends to zero as $t \to \infty$. In other words, a zero solution is a global attractor of the system (3.9) with $\varphi(\sigma) = k\sigma$).

If at the points of differentiability of $\varphi(\sigma)$ the condition

$$\mu_1 < \varphi'(\sigma) < \mu_2 \tag{3.22}$$

is satisfied, then the system (3.9) is stable in the large.

The Kalman conjecture is a strengthening of the Aizerman conjecture, where in place of the condition (3.22) on the derivative of nonlinearity it is required that the nonlinearity itself belongs to a linear sector.

To justify the method of harmonic balance in this *critical case* special nonlinearities will be considered. Let us assume first that $\mu_1 = 0$, $\mu_2 > 0$ and consider the system (3.12) with nonlinearity $\varphi^0(\sigma)$ of a special form

$$\varphi^0(\sigma) = \begin{cases} \mu\sigma, & \forall |\sigma| \le \varepsilon; \\ \operatorname{sign}(\sigma) M\varepsilon^3, & \forall |\sigma| > \varepsilon. \end{cases} \tag{3.23}$$

Here $\mu < \mu_2$ and M are certain positive numbers and ε is a small positive parameter.

Then the following result is valid.

Theorem 3.2 ([6, 16]) *If the inequalities*

$$b_1 < 0, \quad 0 < \mu b_2 \omega_0 \left(\mathbf{c}_3^* \mathbf{b}_3 + b_1\right) + b_1 \omega_0^2$$

are satisfied, then for small enough ε the system (3.12) *with nonlinearity* (3.23) *has an orbitally stable periodic solution*

$$\begin{aligned} y_1(t) &= -\sin(\omega_0 t) y_2(0) + O(\varepsilon), \\ y_2(t) &= \cos(\omega_0 t) y_2(0) + O(\varepsilon), \\ \mathbf{y}_3(t) &= \mathbf{O}_{n-2}(\varepsilon) \end{aligned} \tag{3.24}$$

with the initial date

$$\begin{aligned} y_1(0) &= O\big(\varepsilon^2\big), \\ y_2(0) &= -\sqrt{\frac{\mu(\mu b_2 \omega_0(\mathbf{c}_3^* \mathbf{b}_3 + b_1) + b_1 \omega_0^2)}{-3\omega_0^2 M b_1}} + O(\varepsilon), \\ \mathbf{y}_3(0) &= \mathbf{O}_{n-2}\big(\varepsilon^2\big). \end{aligned} \tag{3.25}$$

The methods for the proof of this theorem are developed in [6, 15, 16, 21].

3.2.2 Hidden Oscillations in Counterexamples to the Aizerman and Kalman Conjectures

Based on this theorem, it is possible to apply the described above multi-step procedure for the localization of hidden oscillations: the initial data, obtained analytically, allows one to step aside from stable zero equilibrium and to start a numerical localization of possible oscillations.

Consider a finite sequence of piecewise-linear functions

$$\varphi^j(\sigma) = \begin{cases} \mu\sigma, & \forall |\sigma| \le \varepsilon_j, \\ \operatorname{sign}(\sigma) M \varepsilon_j^3, & \forall |\sigma| > \varepsilon_j, \end{cases} \qquad \varepsilon_j = \frac{j}{m}\sqrt{\frac{\mu}{M}} \quad j = 1, \dots, m. \tag{3.26}$$

Here the function $\varphi^m(\sigma)$ is a monotone continuous piecewise-linear function $\operatorname{sat}(\sigma)$ ("saturation"). Choose m in such a way that the graphs of the functions φ^j and φ^{j+1} are slightly distinct from each other outside small neighborhoods of points of discontinuity.

Suppose that the periodic solution $\mathbf{x}^m(t)$ of the system (3.9) with the monotone and continuous function $\varphi^m(\sigma) = \operatorname{sat}(\sigma)$ is computed. In this case a similar computational procedure for the sequence of systems with nonlinearities can be organized:

$$\begin{aligned} \theta^i(\sigma) &= \varphi^m(\sigma) + \operatorname{sat}(\sigma) + \frac{i}{10}\big(\tanh(\sigma) - \operatorname{sat}(\sigma)\big), \quad i = 0, \dots, 10, \\ \theta^0(\sigma) &= \operatorname{sat}(\sigma), \qquad \theta^{10}(\sigma) = \tanh(\sigma) = \frac{e^{\sigma} - e^{-\sigma}}{e^{\sigma} + e^{-\sigma}}. \end{aligned} \tag{3.27}$$

Note that, using the similar technique of small changes, it is also possible to approach other continuous monotonic increasing functions [26]. The finding of periodic solutions for a system with nonlinearity (3.27) gives a certain counterexample to the Kalman conjecture for each $i = 1, \ldots, 10$.

Consider the following system:

$$\begin{aligned} \dot{x}_1 &= -x_2 - 10\varphi(\sigma), \\ \dot{x}_2 &= x_1 - 10.1\varphi(\sigma), \\ \dot{x}_3 &= x_4, \\ \dot{x}_4 &= -x_3 - x_4 + \varphi(\sigma), \\ \sigma &= x_1 - 10.1x_3 - 0.1x_4. \end{aligned} \tag{3.28}$$

Here for $\varphi(\sigma) = k\sigma$ the linear system (3.28) is stable for $k \in (0, 9.9)$ (see (3.25)). For piecewise-continuous nonlinearity $\varphi(\sigma) = \varphi^0(\sigma)$ with sufficiently small ε there exists a periodic solution.

Now let us use the algorithm for construction of periodic solutions. Suppose $\mu = M = 1$, $\varepsilon_1 = 0.1$, $\varepsilon_2 = 0.2, \ldots, \varepsilon_{10} = 1$. For $j = 1, \ldots, 10$, the solutions of the system (3.28) with nonlinearity $\varphi(\sigma)$ equal to $\varphi^j(\sigma)$ can be constructed sequentially. Here for all ε_j, $j = 1, \ldots, 10$ there exists a periodic solution.

At the first step, for $j = 0$, the initial data of stable periodic oscillation take the form

$$\begin{aligned} x_1(0) &= O(\varepsilon), & x_3(0) &= O(\varepsilon), \\ x_2(0) &= -1.7513 + O(\varepsilon) & x_4(0) &= O(\varepsilon). \end{aligned} \tag{3.29}$$

Therefore for $j = 1$ a trajectory starts from the point $x_1(0) = x_3(0) = x_4(0) = 0$, $x_2(0) = -1.7513$. The projection of this trajectory on the plane (x_1, x_2) and the sector of linear stability are shown in Fig. 3.6 for the odd steps.

From Fig. 3.6 it follows that at each step after a transient process a stable periodic solution is reached. At each step, the last trajectory point is used as the initial data for the next step of the computational procedure.

Proceeding this procedure for $j = 3, \ldots, 10$, one sequentially approximates a periodic solution of the initial system (3.28) (Fig. 3.7). It should also be noted that if in place of sequential increasing ε_j to compute, for $\varepsilon = 1$, a solution with the initial data according to (3.29), then the solution will "fall down" to zero.

Change the nonlinearity $\varphi(\sigma)$ to the increasing function $\theta^i(\sigma)$, and continue sequential construction of periodic solutions of the system (3.28) for $i = 1, \ldots, 10$. The obtained periodic solutions are shown in Fig. 3.8.

At the last step for the system (3.28) with smooth strictly increasing nonlinearity

$$\varphi(\sigma) = \tanh(\sigma) = \frac{e^{\sigma} - e^{-\sigma}}{e^{\sigma} + e^{-\sigma}}, \quad 0 < \frac{d}{d\sigma}\tanh(\sigma) \le 1, \ \forall\sigma \tag{3.30}$$

there exists a periodic solution (Fig. 3.9).

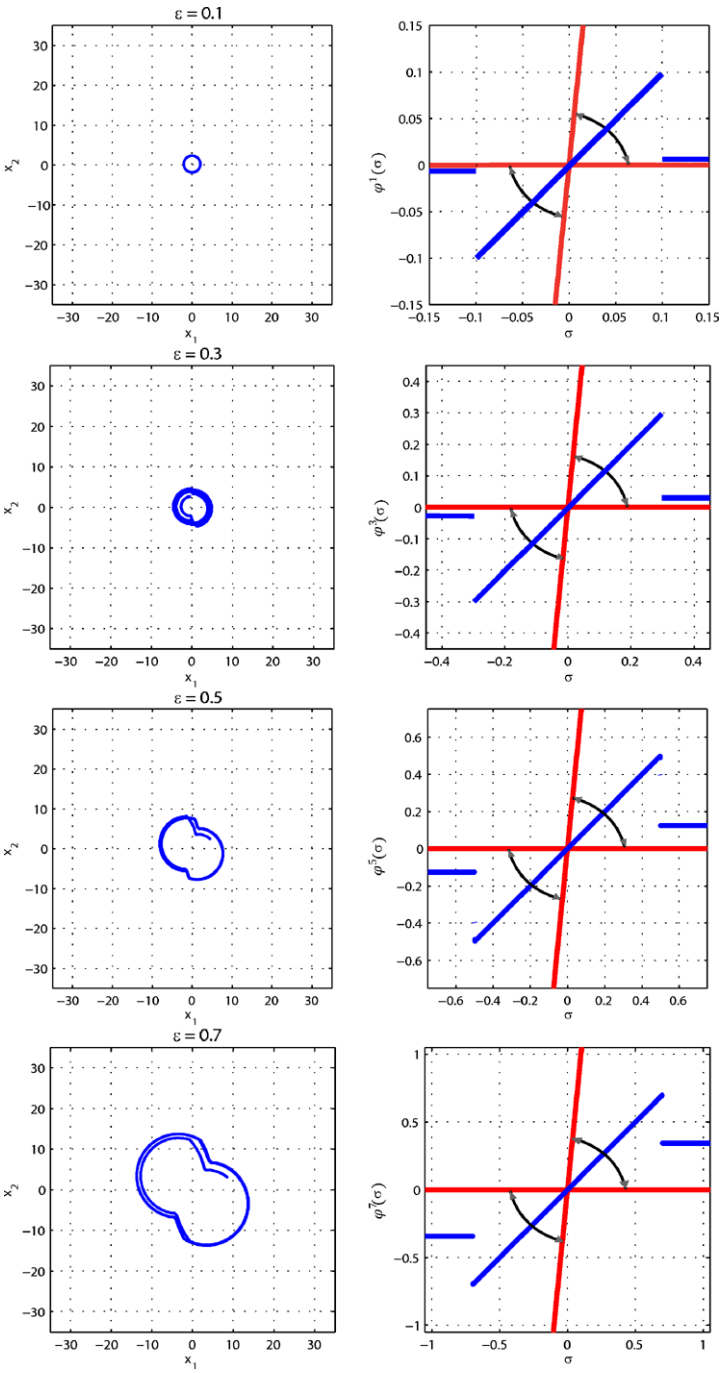


Fig. 3.6 ε_j: trajectory projection on the plane (x_1, x_2) and nonlinearity

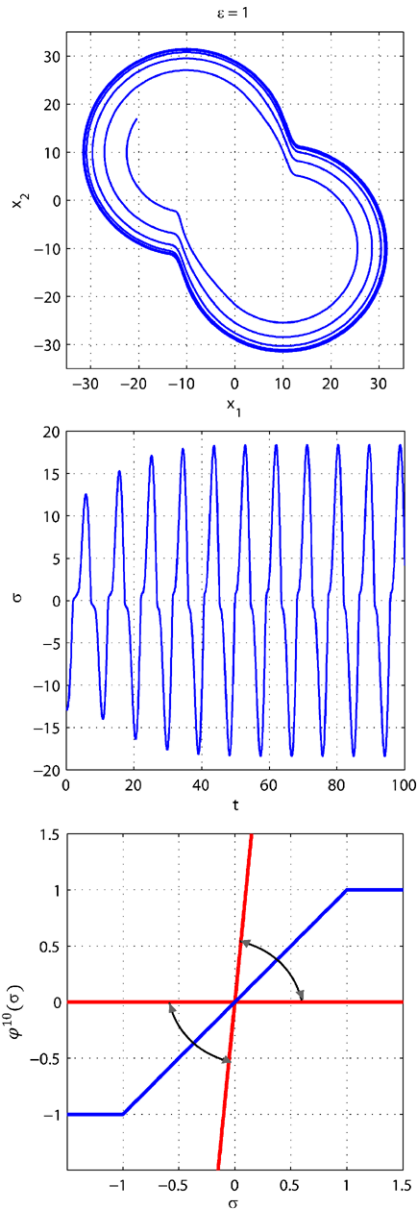


Fig. 3.7 Hidden oscillation projection on the plane (x_1, x_2), the system output $\sigma(t)$, and nonlinearity

3.2.3 Hidden Chaotic Attractors in the Chua Circuit

The development of modern computers allows one to perform numerical simulation of nonlinear chaotic systems and to obtain new information on the structure of their trajectories. In the well-known Lorenz, Chen, Chua, and many other chaotic

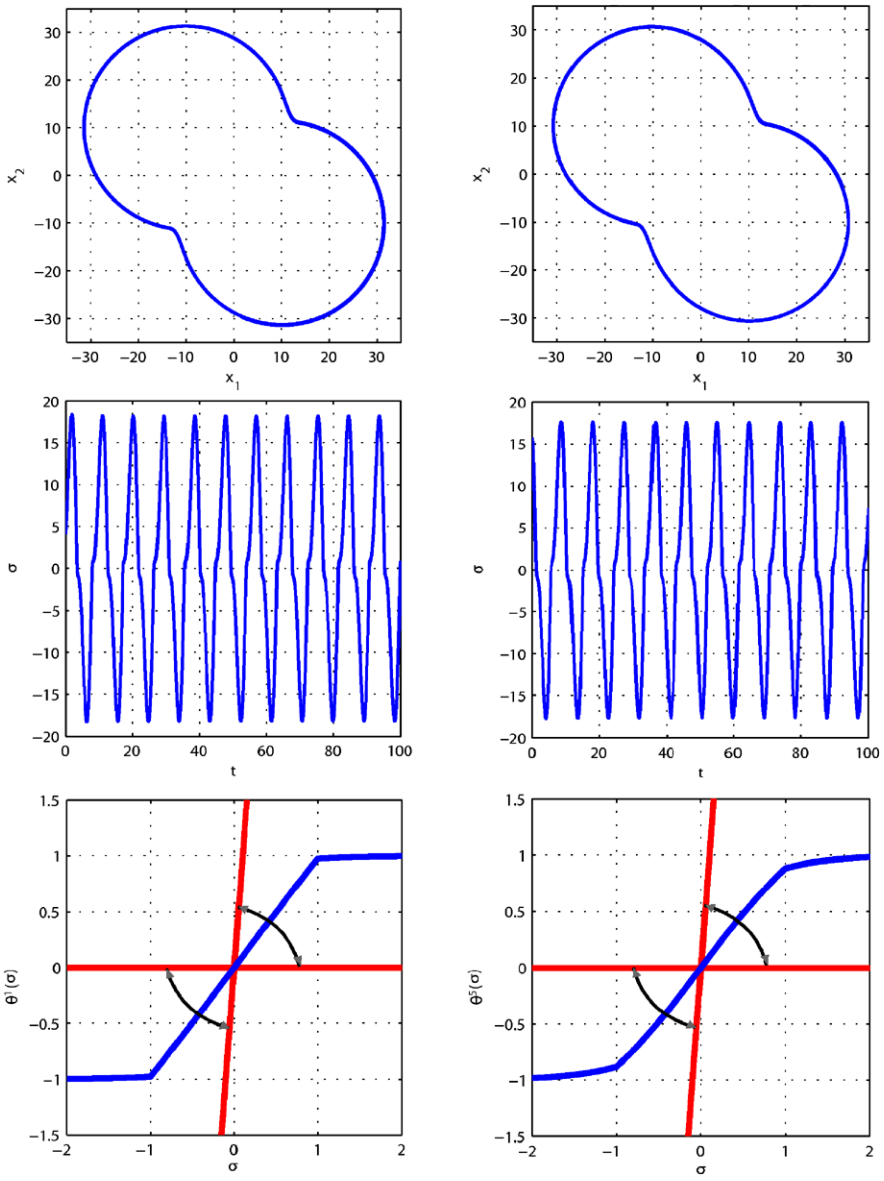


Fig. 3.8 Trajectory projection on the plane (x_1, x_2), system output and nonlinearity

dynamical systems the classical attractors are self-exciting attractors and can be obtained numerically by means of a standard computational procedure. In contradiction, there are attractors of another type: *hidden chaotic attractors*, which cannot be obtained by a standard computational procedure and show limitations of such a simple computational approach.

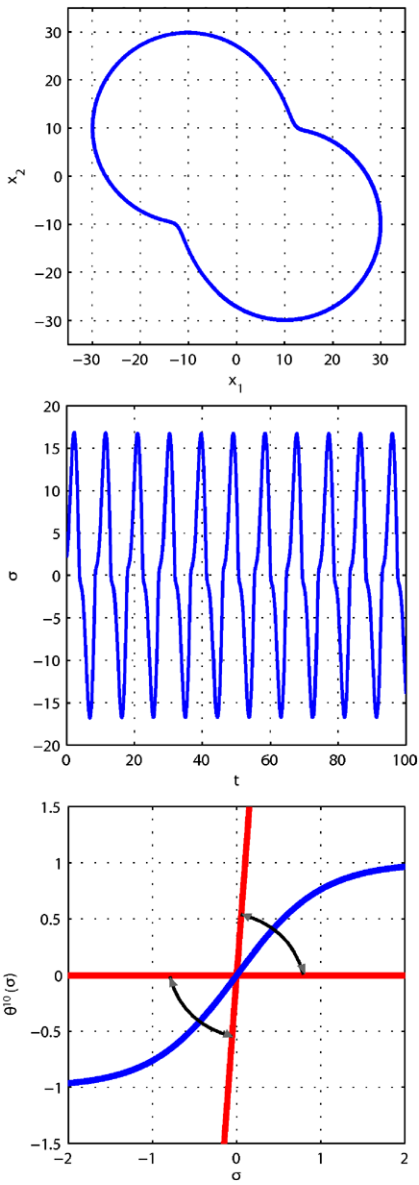


Fig. 3.9 A counterexample to the Kalman conjecture: hidden oscillation in a system with the increasing nonlinearity $\tanh(\sigma)$, which belongs to the sector of linear stability

In 2010, for the first time, a chaotic hidden attractor was discovered [14, 23] in the Chua circuit, which is described by a three-dimensional dynamical system. Let us demonstrate the application of the above algorithm for localization of a hidden chaotic attractor in the Chua system. For this purpose, rewrite the Chua system (3.4) as (3.7)

$$\frac{d\mathbf{x}}{dt} = \mathbf{P}\mathbf{x} + \mathbf{q}\psi\left(\mathbf{r}^*\mathbf{x}\right), \quad \mathbf{x} \in \mathbb{R}^3. \tag{3.31}$$

Here

$$\mathbf{P}=\begin{pmatrix}-\alpha(m_1+1) & \alpha & 0\\ 1 & -1 & 1\\ 0 & -\beta & -\gamma\end{pmatrix},\qquad \mathbf{q}=\begin{pmatrix}-\alpha\\ 0\\ 0\end{pmatrix},\qquad \mathbf{r}=\begin{pmatrix}1\\ 0\\ 0\end{pmatrix},$$

$$\psi(\sigma)=(m_0-m_1)\mathrm{sat}(\sigma).$$

Introduce the coefficient k and the small parameter ε, and represent the system (3.31) as (3.10), namely

$$\frac{d\mathbf{x}}{dt}=\mathbf{P_0}\mathbf{x}+\mathbf{q}\varepsilon\varphi\big(\mathbf{r}^*\mathbf{x}\big), \tag{3.32}$$

where

$$\mathbf{P_0}=\mathbf{P}+k\mathbf{q}\mathbf{r}^*=\begin{pmatrix}-\alpha(m_1+1+k) & \alpha & 0\\ 1 & -1 & 1\\ 0 & -\beta & -\gamma\end{pmatrix},$$

$$\lambda_{1,2}^{\mathbf{P}_0}=\pm i\omega_0,\qquad \lambda_3^{\mathbf{P}_0}=-d,$$

$$\varphi(\sigma)=\psi(\sigma)-k\sigma=(m_0-m_1)\mathrm{sat}(\sigma)-k\sigma.$$

By the nonsingular linear transformation $\mathbf{x}=\mathbf{S}\mathbf{y}$ the system (3.32) is reduced to the form (3.12), namely

$$\frac{d\mathbf{y}}{dt}=\mathbf{A}\mathbf{y}+\mathbf{b}\varepsilon\varphi\big(\mathbf{c}^*\mathbf{y}\big), \tag{3.33}$$

where

$$\mathbf{A}=\begin{pmatrix}0 & -\omega_0 & 0\\ \omega_0 & 0 & 0\\ 0 & 0 & -d\end{pmatrix},\qquad \mathbf{b}=\begin{pmatrix}b_1\\ b_2\\ 1\end{pmatrix},\qquad \mathbf{c}=\begin{pmatrix}1\\ 0\\ -h\end{pmatrix}.$$

The transfer function $W_{\mathbf{A}}(p)$ of the system (3.33) can be represented as

$$W_{\mathbf{A}}(p)=\frac{-b_1p+b_2\omega_0}{p^2+\omega_0^2}+\frac{h}{p+d}.$$

Further, using the equality of the transfer functions of the systems (3.32) and (3.33), one obtains

$$W_{\mathbf{A}}(p)=\mathbf{r}^*(\mathbf{P}_0-p\mathbf{I})^{-1}\mathbf{q}.$$

This implies the following relations:

$$\begin{aligned} k &= \frac{-\alpha(m_1 + m_1\gamma + \gamma) + \omega_0^2 - \gamma - \beta}{\alpha(1+\gamma)}, \\ d &= \frac{\alpha + \omega_0^2 - \beta + 1 + \gamma + \gamma^2}{1+\gamma}, \\ h &= \frac{\alpha(\gamma + \beta - (1+\gamma)d + d^2)}{\omega_0^2 + d^2}, \\ b_1 &= \frac{\alpha(\gamma + \beta - \omega_0^2 - (1+\gamma)d)}{\omega_0^2 + d^2}, \\ b_2 &= \frac{\alpha((1+\gamma-d)\omega_0^2 + (\gamma+\beta)d)}{\omega_0(\omega_0^2 + d^2)}. \end{aligned} \tag{3.34}$$

Since the system (3.32) can be reduced to the form (3.33) by the nonsingular linear transformation $\mathbf{x} = \mathbf{S}\mathbf{y}$, for the matrix $\mathbf{S}$ the relations

$$\mathbf{A} = \mathbf{S}^{-1}\mathbf{P_{0S}}, \qquad \mathbf{b} = \mathbf{S}^{-1}\mathbf{q}, \qquad \mathbf{c}^* = \mathbf{r}^*\mathbf{S} \tag{3.35}$$

are valid. Having solved these matrix equations, one obtains the transformation matrix

$$\mathbf{S} = \begin{pmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{pmatrix}.$$

Here

$$\begin{gathered} s_{11} = 1, \qquad s_{12} = 0, \qquad s_{13} = -h, \\ s_{21} = m_1 + 1 + k, \qquad s_{22} = -\frac{\omega_0}{\alpha}, \qquad s_{23} = -\frac{h(\alpha(m_1 + 1 + k) - d)}{\alpha}, \\ s_{31} = \frac{\alpha(m_1 + k) - \omega_0^2}{\alpha}, \qquad s_{32} = -\frac{\alpha(\beta+\gamma)(m_1+k) + \alpha\beta - \gamma\omega_0^2}{\alpha\omega_0}, \\ s_{33} = h\frac{\alpha(m_1+k)(d-1) + d(1+\alpha-d)}{\alpha}. \end{gathered}$$

By (3.19), for small enough ε initial data for the first step of multistage localization procedure take the form

$$\mathbf{x}(0) = \mathbf{S}\mathbf{y}(0) = \mathbf{S}\begin{pmatrix} a_0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a_0 s_{11} \\ a_0 s_{21} \\ a_0 s_{31} \end{pmatrix}.$$

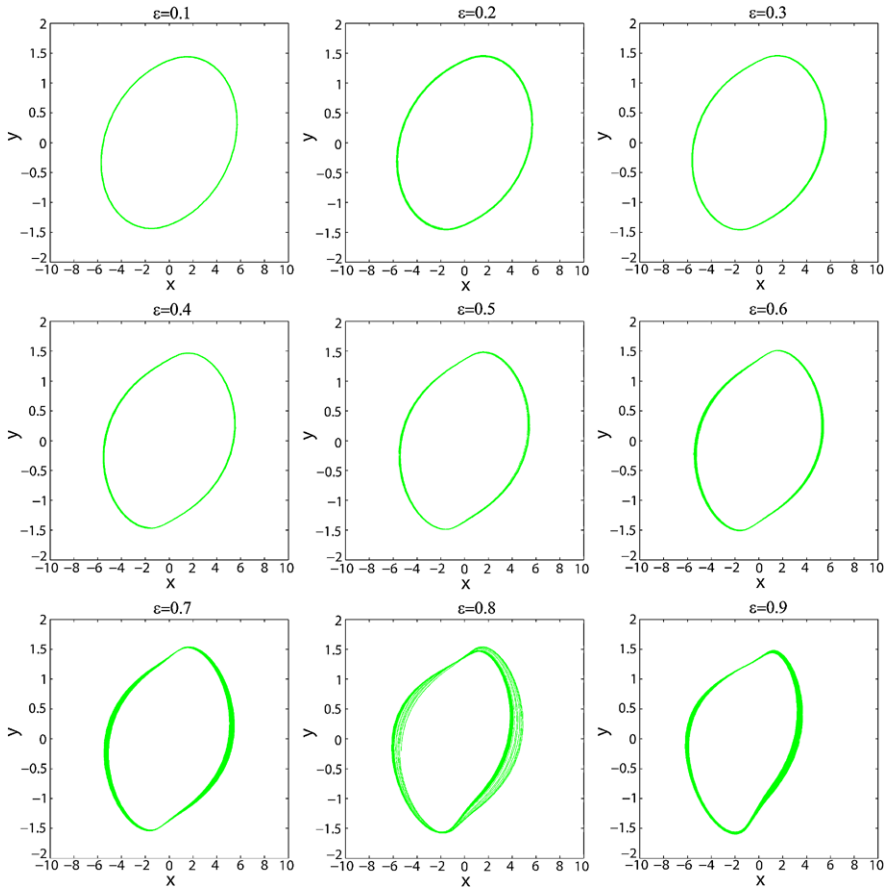


Fig. 3.10 Localization of a hidden chaotic attractor: a road to chaos: the projections of trajectories on the plane (x, y)

Returning to the Chua system's denotations, for determining the initial data of the starting solution of the multistage procedure we have the following formulas:

$$x(0) = a_0, \qquad y(0) = a_0(m_1 + 1 + k), \qquad z(0) = a_0 \frac{\alpha(m_1 + k) - \omega_0^2}{\alpha}. \quad (3.36)$$

Consider the system (3.32) with the parameters

$$\alpha = 8.4562, \qquad \beta = 12.0732, \qquad \gamma = 0.0052,$$
$$m_0 = -0.1768, \qquad m_1 = -1.1468. \quad (3.37)$$

Note that in this case for the considered values of parameters there are three equilibria in the system: a locally stable zero equilibrium and two saddle equilibria.

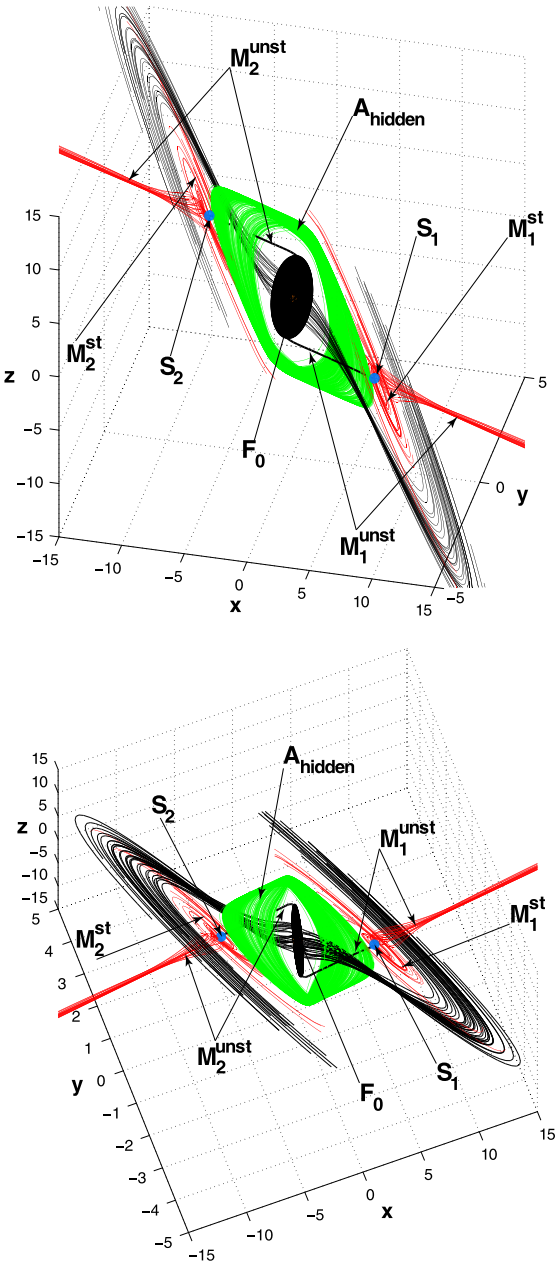


Fig. 3.11 Equilibrium, stable manifolds of saddles, and localization of hidden attractor

Now we apply the above procedure of hidden attractors' localization to the Chua system (3.31) with the parameters (3.37). For this purpose let us compute a starting frequency and a coefficient of harmonic linearization. We have

$$\omega_0 = 2.0392, \qquad k = 0.2098.$$

Then, one computes solutions of the system (3.32) with nonlinearity $\varepsilon\varphi(x) = \varepsilon(\psi(x) - kx)$, sequentially increasing ε from the value $\varepsilon_1 = 0.1$ to $\varepsilon_{10} = 1$ with the step 0.1 (see Fig. 3.10).

By (3.34) and (3.36) we obtain the initial data

$$x(0) = 9.4287, \qquad y(0) = 0.5945, \qquad z(0) = -13.4705$$

for the first step of the multistage procedure for the construction of solutions. For the value of the parameter $\varepsilon_1 = 0.1$, after the transient process the computational procedure reaches the starting oscillation $\mathbf{x}^1(t)$. Further, by the sequential transformation of $\mathbf{x}^j(t)$ with increasing the parameter ε_j, using the numerical procedure, for the original Chua system (3.31) the set $\mathscr{A}_{\text{hidden}}$ is computed. This set is presented in Fig. 3.11.

It should be noted that the decreasing of the integration step, the increasing of integration time, and the computation of different trajectories of the original system with initial data from a small neighborhood of $\mathscr{A}_{\text{hidden}}$ lead to the localization of the same set $\mathscr{A}_{\text{hidden}}$ (all the computed trajectories densely trace the set $\mathscr{A}_{\text{hidden}}$). Note also that for the computed trajectories Zhukovsky instability and the positiveness of the Lyapunov exponent [19] is observed.

The behavior of system trajectories in the neighborhood of equilibria is presented in Fig. 3.11. Here $M^{\text{unst}}_{1,2}$ are unstable manifolds, $M^{\text{st}}_{1,2}$ are stable manifolds. Thus, in a phase space of the system there are stable separating manifolds of saddles.

The above and the remark on the existence, in the system, of a locally stable zero equilibrium F_0, attracting the stable manifolds $M^{\text{st}}_{1,2}$ of two symmetric saddles S_1 and S_2, lead to the conclusion that in $\mathscr{A}_{\text{hidden}}$ a hidden strange attractor is computed.

3.3 Conclusion

The study of hidden oscillations and hidden chaotic attractors requires the development of new analytical and numerical methods. This survey includes discussion on new analytical-numerical approaches to investigation of hidden oscillations in dynamical systems, based on the development of numerical methods, computers, and an applied bifurcation theory, which suggests revising early ideas on the application of the small parameter method and the harmonic linearization [16, 18, 23, 26].

Acknowledgements This work was partly supported by Academy of Finland, Ministry of Education and Science of the Russian Federation (Federal target program "Scientific and Scientific-Pedagogical Cadres for Innovative Russia" for the years 2009–2013), Russian Foundation for Basic Research and Saint-Petersburg State University.

References

1. Aizerman MA (1949) On a problem concerning the stability "in the large" of dynamical systems. Usp Mat Nauk 4(4):187–188. In Russian

2. Andronov AA, Vitt AA, Khaikin SE (1966) Theory of oscillators. Pergamon Press, Oxford
3. Belousov BP (1959) A periodic reaction and its mechanism. In: Collection of abstracts on radiation medicine, 1958. Medgiz, Moscow, pp 145–147
4. Bernat J, Llibre J (1996) Counterexample to Kalman and Markus-Yamabe conjectures in dimension larger than 3. Dyn Contin Discrete Impuls Syst 2(3):337–379
5. Bilotta E, Pantano P (2008) A gallery of Chua attractors. World Scientific series on nonlinear science. Series A: monographs and treatises, vol 61. World Scientific, Hackensack, NJ
6. Bragin VO, Vagaitsev VI, Kuznetsov NV, Leonov GA (2011) Algorithms for finding hidden oscillations in nonlinear systems. The Aizerman and Kalman conjectures and Chua's circuits. J Comput Syst Sci Int 50(4):511–543
7. Chua LO (1992) A zoo of strange attractors from the canonical Chua's circuits. In: Proceedings of the 35th midwest symposium on circuits and systems, vol 2. IEEE, New York, pp 916–926
8. Gubar' NA (1961) Investigation of a piecewise linear dynamical system with three parameters. J Appl Math Mech 25(6):1519–1535
9. Kalman RE (1981) Physical and mathematical mechanisms of instability in nonlinear automatic control systems. Trans Am Soc Mech Eng 79(3):553–566
10. Khalil HK (2002) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River
11. Krylov AN (1936) Vibration of ships. Gl Red Sudostroit Lit, Moscow. In Russian
12. Kuznetsov NV, Kuznetsova OA, Leonov GA (2011) Investigation of limit cycles in two-dimensional quadratic systems. In: Proceedings of the 2nd international symposium rare attractors and rare phenomena in nonlinear dynamics (RA'11), pp 120–123
13. Kuznetsov NV, Leonov GA (2008) Lyapunov quantities, limit cycles and strange behavior of trajectories in two-dimensional quadratic systems. J Vibroeng 10(4):460–467
14. Kuznetsov NV, Leonov GA, Vagaitsev VI (2010) Analytical-numerical method for attractor localization of generalized Chua's system. In: Periodic control systems, vol 4, Part 1. IFAC
15. Leonov GA (2009) On a harmonic linearization method. Dokl Math 79(1):144–146
16. Leonov GA (2010) Effective methods in the search for periodic oscillations in dynamical systems. J Appl Math Mech 74(1):24–50
17. Leonov GA (2011) Four normal size limit cycles in two-dimensional quadratic systems. Int J Bifurc Chaos Appl Sci Eng 21(2):425–429
18. Leonov GA, Bragin VO, Kuznetsov NV (2010) Algorithm for constructing counterexamples to the Kalman problem. Dokl Math 82(1):540–542
19. Leonov GA, Kuznetsov NV (2007) Time-varying linearization and the Perron effects. Int J Bifurc Chaos Appl Sci Eng 17(4):1079–1107
20. Leonov GA, Kuznetsov NV (2010) Limit cycles of quadratic systems with a perturbed third-order focus and a saddle equilibrium state at infinity. Dokl Math 82(2):693–696
21. Leonov GA, Kuznetsov NV (2011) Algorithms for searching for hidden oscillations in the Aizerman and Kalman problems. Dokl Math 84(1):475–481
22. Leonov GA, Kuznetsov NV, Kudryashova EV (2011) A direct method for calculating Lyapunov quantities of two-dimensional dynamical systems. Proc Steklov Inst Math 272(Suppl 1):119–126
23. Leonov GA, Kuznetsov NV, Vagaitsev VI (2011) Localization of hidden Chua's attractors. Phys Lett A 375(23):2230–2233
24. Leonov GA, Kuznetsova OA (2009) Evaluation of the first five Lyapunov exponents for the Liénard system. Dokl Phys 54(3):131–133
25. Leonov GA, Kuznetsova OA (2010) Lyapunov quantities and limit cycles of two-dimensional dynamical systems. Analytical methods and symbolic computation. Regul Chaotic Dyn 15(2–3):354–377
26. Leonov GA, Vagaitsev VI, Kuznetsov NV (2010) Algorithm for localizing Chua attractors based on the harmonic linearization method. Dokl Math 82(1):663–666
27. Lorenz EN (1963) Deterministic nonperiodic flow. J Atmos Sci 20(2):130–141
28. Markus L, Yamabe H (1960) Global stability criteria for differential systems. Osaka Math J 12:305–317

29. Matsumoto T (1984) A chaotic attractor from Chua's circuit. IEEE Trans Circuits Syst 31(12):1055–1058
30. Pliss VA (1958) Some problems in the theory of the stability of motion. Izd LGU, Leningrad
31. Rayleigh JWS (1877) The theory of sound. MacMillan, London
32. Stoker JJ (1950) Nonlinear vibrations in mechanical and electrical systems. Interscience, New York
33. Timoshenko S (1928) Vibration problems in engineering. Van Nostrand, New York
34. van der Pol B (1926) On "relaxation-oscillations". Philos Mag Ser 7 2(11):978–992

Chapter 4
Numerical Study of a High Order 3D FEM-Level Set Approach for Immiscible Flow Simulation

Stefan Turek, Otto Mierka, Shuren Hysing, and Dmitri Kuzmin

Abstract Numerical simulation of incompressible multiphase flows with immiscible fluids is still a challenging field, particularly for 3D configurations undergoing complex topological changes. In this paper, we discuss a 3D FEM approach with high-order Stokes elements (Q_2/P_1) for velocity and pressure on general hexahedral meshes. A discontinuous Galerkin approach with piecewise linear polynomials (dG(1)) is used to treat the Level Set function. The developed methodology allows the application of special redistancing algorithms which do not change the position of the interface. We explain the corresponding FEM techniques for treating the advection steps and surface tension effects, and validate the corresponding 3D code with respect to both numerical test cases and experimental data. The corresponding applications describe the classical rising bubble problem for various parameters and the generation of droplets from a viscous liquid jet in a coflowing surrounding fluid. Both of these applications can be used for rigorous benchmarking of 3D multiphase flow simulations.

Keywords Multiphase flow · Finite elements · Discontinuous Galerkin · Numerical simulation

S. Turek (✉) · O. Mierka
Institut für Angewandte Mathematik, TU Dortmund, Vogelpothsweg 87, 44227 Dortmund, Germany
e-mail: ture@featflow.de

O. Mierka
e-mail: omierka@math.uni-dormtund.de

S. Hysing
Department of Mathematics, Shanghai Jiaotong University, Shanghai, China
e-mail: shuren.hysing@sjtu.edu.cn

D. Kuzmin
Lehrstuhl für Angewandte Mathematik III, Universität Erlangen-Nürnberg, Cauerstraße 11, 91058 Erlangen, Germany
e-mail: kuzmin@am.uni-erlangen.de

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_4,

4.1 Introduction

Multiphase flow problems are very important in many applications, and performing accurate, robust and efficient numerical simulations of them has been the object of numerous research and simulation projects for several years. One of the main challenges for the underlying numerical methods is that the position of the moving interface between two fluids is unknown and must be determined as a part of the boundary value problem which should be solved. If we assume a domain Ω with two immiscible fluids, then the time-dependent subdomains $\Omega_1(t)$ and $\Omega_2(t)$ are bounded by an external boundary Σ and a dynamic interior boundary or interface $\Gamma(t)$ which might consist of several components (see Fig. 4.1).

Then, the usual model for laminar (multiphase) flow is described by the incompressible Navier-Stokes equations

$$\rho(\mathbf{x})\left[\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right]-\nabla\cdot\left(\mu(\mathbf{x})\left[\nabla\mathbf{u}+(\nabla\mathbf{u})^T\right]\right)+\nabla p=\rho(\mathbf{x})\mathbf{g}+\mathbf{f}_\Gamma(\sigma), \quad (4.1)$$

$$\nabla\cdot\mathbf{u}=0 \quad \text{in } \Omega=\Omega_1\cup\Gamma\cup\Omega_2, \quad (4.2)$$

which contain an additional force term $\mathbf{f}_\Gamma(\sigma)$ due to the surface tension σ at the free interface Γ. Here, the density ρ as well as the viscosity μ are variable and discontinuous, that is

$$\rho(\mathbf{x},t)=\begin{cases}\rho_1, & \forall\mathbf{x}\in\Omega_1(t),\\ \rho_2, & \forall\mathbf{x}\in\Omega_2(t),\end{cases}\qquad \mu(\mathbf{x},t)=\begin{cases}\mu_1, & \forall\mathbf{x}\in\Omega_1(t),\\ \mu_2, & \forall\mathbf{x}\in\Omega_2(t),\end{cases} \quad (4.3)$$

which significantly influences the velocity $\mathbf{u}$ as well as the pressure p.

This contribution describes the numerical analysis and application of a new Level Set approach in the framework of the Finite Element Method (FEM) for such multiphase flow problems. For this reason the open-source CFD package FEATFLOW (www.featflow.de) was utilized and extended with the corresponding newly created Level Set module so that the existing methodology of the FEATFLOW approach, namely flexible, high order FEM discretization schemes in space and time with flux correction [34] and edge-oriented stabilization techniques [62], unstructured meshes with adaptive grid deformation, efficient Newton-Multigrid solvers, and parallelization based on domain decomposition could be directly exploited.

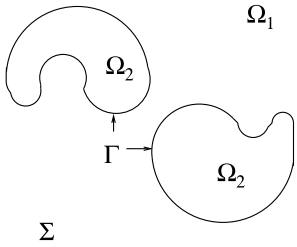


Fig. 4.1 A sketch of the complete domain $\Omega=\Omega_1\cup\Gamma\cup\Omega_2$

The outline of the paper is as follows: after a short description in Sect. 4.2 of the state-of-the-art regarding interface tracking and capturing methods, particularly for Level Set approaches, we describe in Sect. 4.3 the chosen solution technique which is based on a discrete projection method [60, 61] for the Navier–Stokes equations, the Level Set advection equation, and the corresponding reinitialization procedure. Moreover, the discretization aspects regarding the incompressible Navier–Stokes equations using the Crank–Nicolson method and the Q_2/P_1 element pair are discussed in Sect. 4.3, too, whereas the details of the employed Discontinuous Galerkin FEM approach with P_1 elements for the Level Set equation can be found in Sect. 4.4. Section 4.5 presents several numerical results which first of all evaluate the grid-independent behaviour of the developed CFD solver.

Furthermore, based on experimental and computational studies, we propose and discuss new benchmark configurations for prototypical 3D multiphase flows which can be used for 'simple' validation and evaluation of multiphase flow CFD codes without the necessity of complex postprocessing operations. Finally, the results are summarized in Sect. 4.6 where an outlook is provided for more complex 3D multiphase flow problems.

4.2 Mathematical Model

The free interface Γ is constantly being deformed and moved so that its position has to be treated as unknown and determined in every time step. Depending on the technique for the representation of the interface, one can distinguish between *front tracking* and *front capturing* approaches which can be realized on fixed as well as dynamic moving meshes. For an overview of existing numerical approaches and their classification, we recommend [52, 56]. The "natural" front tracking approach [21, 41, 55, 65] is based on an explicit tracing of the dynamic interface between the two phases. Here, in the case of Lagrangian finite element methods [25], the underlying mesh has to be constantly adapted to the free interface so that the grid points move with the interface. More flexibility is promised by the *Arbitrary Lagrangian Eulerian* (ALE) formulation [1, 2, 7, 17, 19, 51] which is based on local grid adaptation and which provides excellent results in the case of moderate deformations (for instance for small waves at the free surface). Moreover, there are many more techniques of fictitious domain and Chimera type which allow the highly accurate tracking of the dynamic interfaces via overlapping surface meshes [26]. However, such front tracking methods do not allow large deformations of the free interfaces or even topological changes such as drop formation and bubble breakup or coalescence, which typically lead to highly distorted meshes. Moreover, the computational costs regarding the implementation and also CPU timings are often very large for complex 3D simulations.

In contrast to such Lagrangian methods, Eulerian front capturing methods are much more robust and flexible. They are applicable even to free interface problems with significant topology changes (breakup of bubbles, fragmentation, coalescence,

etc.). Based on the early Marker-and-Cell method of Harlow and Welch [67], the implicit reconstruction of the interface is based on an indicator function $\phi(\mathbf{x}, t)$ which contains the information about the corresponding subdomain for the point $\mathbf{x}$ at time t. The distribution in the complete domain Ω can then be calculated via the scalar transport equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \tag{4.4}$$

so that the exact position of the free interface $\Gamma(\phi)$ at any time can be reconstructed from ϕ with the help of postprocessing techniques. One of the most well-known methods is the Volume-of-Fluid (VOF) method [42, 54] in which case the indicator function ϕ can be interpreted as volume fraction which should have the discrete values 0 or 1 depending on the location of $\mathbf{x}$:

$$\phi(\mathbf{x}, t) = \begin{cases} 1, & \forall \mathbf{x} \in \Omega_1(t), \\ 0, & \forall \mathbf{x} \in \Omega_2(t). \end{cases} \tag{4.5}$$

The numerical drawback of this approach is that artificial diffusion smears out the (originally) discontinuous indicator function which arises from the solution of the discretized advection problems resulting in a boundary layer with $0 < \phi < 1$. Therefore, numerical schemes and locally adapted meshes have to be designed to address this boundary layer as thin as possible so that the corresponding error for reconstructing the free interface is reduced. Moreover, due to the steep gradients and the discontinuity of the indicator function, standard Galerkin schemes lead to unphysical oscillations which significantly deteriorate the accuracy or even lead to unphysical over- and undershoots. As a conclusion, the development of corresponding high-order monotone discretization schemes in combination with unstructured, locally refined meshes still belongs to the numerical challenges one has to solve.

As a successful alternative, the Level Set approach [43, 44, 53] has been established which represents the interface as zero isoline of a continuous indicator function ϕ which should be close to the distance with respect to the free interface

$$\phi(\mathbf{x}, t) = \begin{cases} \operatorname{dist}(\mathbf{x}, \Gamma), & \forall \mathbf{x} \in \Omega_1(t), \\ -\operatorname{dist}(\mathbf{x}, \Gamma), & \forall \mathbf{x} \in \Omega_2(t) \end{cases} \tag{4.6}$$

so that $\Gamma(t) = \{\mathbf{x} \in \Omega \mid \phi(\mathbf{x}, t) = 0\}$ holds. In contrast to the VOF approach, ϕ as a distance function is smooth and allows the calculation of a globally defined normal vector $\mathbf{n}$ towards the interface Γ and of the corresponding curvature via

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}, \qquad \kappa = -\nabla \cdot \mathbf{n} = -\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right). \tag{4.7}$$

Here, special FEM techniques for gradient recovery can be used which allow highly accurate approximations of normals and curvature [56] which are necessary

for the direct evaluation of the surface tension force $\mathbf{f}_\Gamma = \kappa\sigma\delta(\phi)\mathbf{n}$, with $\delta(\phi)$ denoting the corresponding Dirac Delta function. Hence, the development and implementation of a typical Level Set approach consists of performing the following sequence of tasks:

- Discretization of the Level Set transport problem (4.4).
- Reinitialisation, resp., redistancing of the Level Set function.
- Additional correction so that mass and volume are preserved (if necessary).
- Calculation of normal vector fields (and curvature if needed) based on ϕ.
- Evaluation of the discontinuous fluid parameters $\rho(\phi)$, $\mu(\phi)$, and of $\mathbf{f}_\Gamma$, with or without reconstruction of Γ.

The above sequence of tasks involves a myriad of different possibilities and choices which inevitably lead to numerous differing solution approaches. This is evident from the rich collection of publications on Level Set methods which also demonstrates the high potential of these methods for a wide range of applications (see for instance the books by Osher [43] and Sethian [53]). However, the resulting quality of the solutions mainly depends on the underlying numerical and computational approaches, and one has to acknowledge the fact that most of the existing Level Set codes are still based on finite differences on uniform Cartesian meshes which are easy to implement. The drawback is that the computational cost typically is quite high since uniform mesh refinement has to be performed to resolve the necessary scales, particularly near the fluidic interfaces, but also due to complicated geometries with small-scale structures. Unstructured meshes are particularly well suited for such approaches which leads us to finite volume and finite element discretization methods which are the most prominent candidates for unstructured simulation approaches. Examples for corresponding approaches in the framework of VOF and Level Set methods can be found in [3, 7, 9, 16, 29, 38, 40, 49]. In many approaches, for example in the *Interface Proximity Adaption Method* of Barth and Sethian [3], the mesh is locally refined near the interface which also is quite easy to find if ϕ is a distance function [38].

Although finite element methods together with locally refined grids seem to possess a very advantageous behaviour for simulation of multiphase flow problems with free interfaces, most existing Level Set codes are still based on finite differences. It is only during the last ten years that FEM codes have been successfully applied for these special CFD problems ([46, 50, 57]; see also [15, 23, 40, 47, 56, 59]). However, there is still a huge potential for improvement if '*optimal*' modern discretization and solution techniques shall be adapted to the special characteristics of FEM-Level Set methods. In constructing a modern Level Set solver it is important to focus on unstructured meshes with local grid refinement strategies for highly nonstationary multiphase flow simulations, and make detailed studies for higher numerical stability. Additionally, stable and accurate discretization of the convective terms (for instance, VOF and Phase-Field methods show very steep gradients near the interface, similarly as Level Set approaches without redistancing), robust treatment of large density differences, and the handling of large surface tension σ also require special attention.

Summarizing the properties of FEM-Level Set techniques for multiphase flow problems, we can conclude the following (potentially) advantageous behaviour in comparison to interface tracking methods as well as VOF and Phase-Field approaches which motivates our recent and future work for the combination of FEM and Level Set methods:

- If the Level Set function satisfies the distance property, it is smooth so that even on highly uniform meshes qualitatively good results can be obtained. Local refinement around the interface will help to improve the accuracy, but in contrast to VOF and Phase-Field methods, which may lead to smeared interfaces due to numerical diffusion or to unphysical oscillations due to steep gradients, adaptive meshes are not necessary.
- Accurate FEM discretizations of a higher order can be adapted to the special characteristics of Level Set functions, that means higher smoothness because of the distance function properties.
- Accurate representations of the interface are provided, without explicit description, but even for complex geometrical changes, which is important for handling the surface tension term.
- Auxiliary quantities like normal vectors and curvature are provided, even globally, which is particularly advantageous for the Continuous Surface Force (CSF) [6] approach.

On the other hand, there are still several problems with Level Set approaches (and some of them are also valid for VOF and Phase-Field methods) which are numerically challenging and which are in the focus of our recent and also planned research activities:

- The standard Level Set formulation is not conservative which may lead to mass loss.
- Since reinitialisation is necessary to preserve the distance property, often highly expensive computational operations might be necessary, for instance via solving globally the Eikonal equation, or redistancing is based on 'cheaper' methods which however change the position and shape of the interface, again leading to mass loss.
- Due to the standard explicit treatment of surface tension, the time step size is restricted by the *capillary time step restriction*, that means the necessary time steps depend by purely numerical reasons on the size of surface tension and on the local mesh size.

In the following sections, we first of all describe the overall solution technique which is based on a discrete projection method which is followed by a discussion of the FEM discretization details, particularly regarding the Discontinuous Galerkin approach for treating the Level Set equation.

4.3 Discrete Projection Methods for Navier–Stokes Equations

In this section, we briefly review the '*Discrete Projection Method*' as a special variant of Multilevel Pressure Schur Complement (MPSC) approaches for the solution of incompressible flow problems, and we combine it with FEM discretization techniques. We will explain some characteristics of high-resolution FEM schemes as applied to incompressible flow problems and discuss the computational details regarding the efficient numerical solution of the resulting nonlinear and linear algebraic systems. Furthermore, we will discuss the coupling mechanisms between the 'basic' flow model (standard Navier–Stokes equations for velocity and pressure) and the scalar transport equations for the Level Set indicator function in our multiphase flow solver.

4.3.1 Discretization Techniques

For a better illustration, we consider first of all numerical solution techniques for the (single phase) incompressible Navier–Stokes equations,

$$\begin{aligned} \mathbf{u}_t - \nu\Delta\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p_\rho &= \mathbf{f}, \\ \nabla\cdot\mathbf{u} = 0, \quad \text{in } \Omega\times(0,T] \quad &\text{with } p_\rho = \frac{p}{\rho} \text{ and } \nu = \frac{\mu}{\rho}, \end{aligned} \tag{4.8}$$

for the given force $\mathbf{f}$ which might contain the surface tension. Moreover, boundary values are prescribed on the boundary $\partial\Omega$ as well as an initial condition at $t = 0$. Solving this problem numerically is still a considerable task in the case of long-time calculations and high Reynolds numbers, particularly in 3D and also in 2D if the time dynamics is complex. The common solution approach is a separate discretization in space and time. We first (semi-) discretize in time by one of the usual methods known from the treatment of ordinary differential equations, such as the Forward or Backward Euler-, the Crank–Nicolson- or Fractional-Step-θ–scheme, or others, and obtain a sequence of generalized stationary Navier-Stokes problems.

Basic θ-scheme Given $\mathbf{u}^n$ and $\Delta t = t_{n+1} - t_n$, then solve for $\mathbf{u} = \mathbf{u}^{n+1}$ and $p_\rho = p_\rho^{n+1}$

$$\frac{\mathbf{u}-\mathbf{u}^n}{\Delta t} + \theta[-\nu\Delta\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u}] + \nabla p_\rho = \mathbf{g}^{n+1}, \qquad \nabla\cdot\mathbf{u} = 0, \quad \text{in } \Omega \tag{4.9}$$

with the right-hand side $\mathbf{g}^{n+1} := \theta\mathbf{f}^{n+1} + (1-\theta)\mathbf{f}^n - (1-\theta)[-\nu\Delta\mathbf{u}^n + \mathbf{u}^n\cdot\nabla\mathbf{u}^n]$.

In the following simulations, the parameter θ is chosen as $\theta = 1/2$, representing the Crank-Nicolson-scheme which is of second order. Alternatively, the Fractional-Step-θ-scheme [63], which uses three different values for θ and for the time step Δt at each time level, is another excellent candidate with slightly better robustness properties.

For the spatial discretization, we choose a finite element approach based on a suitable variational formulation. On the finite mesh $\mathcal{T}_h$ (3D hexahedral elements in our case) covering the domain Ω with the local mesh size h, one defines polynomial trial functions for velocity and pressure. These spaces H_h and L_h should lead to numerically stable approximations as $h \to 0$, i.e., they should satisfy the so-called *inf-sup* (LBB) condition [20]

$$\min_{q_h \in L_h} \max_{\mathbf{v}_h \in H_h} \frac{(q_h, \nabla \cdot \mathbf{v}_h)}{\|q_h\|_0 \, \|\nabla \mathbf{v}_h\|_0} \geq \gamma > 0 \tag{4.10}$$

with a mesh-independent constant γ. While the original FEATFLOW solvers are based on *rotated multilinear* nonconforming finite element functions for the velocity and piecewise constant pressure approximations, we recently extended the complete solver package to higher-order Stokes elements, namely conforming triquadratic ansatz functions for the velocity and linear pressure approximations (Q_2/P_1), which belong to the 'best' finite element pairs for laminar incompressible flow due to their accuracy and robustness. Since so far most of our numerical simulations have been performed for small up to moderate Reynolds numbers, the (nonlinear) convective operator was discretized using standard stabilization techniques only. Currently, we use edge-, resp., face-oriented FEM stabilization techniques [62] which can be easily realized for higher-order ansatz functions, too. Here, special jump terms of the gradient of the solution as well as of the test function have to be included into the weak formulation which leads to a consistent stabilization, for stationary as well as nonstationary configurations. It is planned to apply this technique in the case of higher Reynolds number flows, too, which will be a subject of our further studies for such multiphase flow problems. For an overview regarding such special FEM stabilization techniques, we refer to [45, 62] and particularly to [10] which contains corresponding results for the Q_2/P_1 approach, too.

4.3.2 Solution Techniques

Using the same notation $\mathbf{u}$ and p_ρ also for the coefficient vectors in the representation of the approximate solution, the discretized Navier-Stokes equations may be written as a coupled (nonlinear) algebraic system of the form: Given $\mathbf{u}^n$ and $\mathbf{f}$, compute $\mathbf{u} = \mathbf{u}^{n+1}$ and $p_\rho = p_\rho^{n+1}$ by solving

$$A\mathbf{u} + \Delta t B p_\rho = \mathbf{g}, \qquad B^T \mathbf{u} = 0, \tag{4.11}$$

where

$$\mathbf{g} = \left[M - \theta_1 \Delta t N\left(\mathbf{u}^n\right)\right]\mathbf{u}^n + \theta_2 \Delta t \mathbf{f}^{n+1} + \theta_3 \Delta t \mathbf{f}^n. \tag{4.12}$$

Here and in the following, we use the more compact form for the diffusive and advective part

$$N(\mathbf{v})\mathbf{u} := -\nu \Delta \mathbf{u} + \mathbf{v} \cdot \nabla \mathbf{u}, \tag{4.13}$$

while M is the (lumped) mass matrix [66], B is the discrete gradient operator, and $-B^T$ is the associated divergence operator. Furthermore,

$$A\mathbf{u} = \big[M - \theta \Delta t N(\mathbf{u})\big]\mathbf{u}, \quad N(\mathbf{u}) = K(\mathbf{u}) + \nu L, \tag{4.14}$$

where L is the discrete Laplacian and $K(\mathbf{u})$ is the nonlinear transport operator incorporating a certain amount of artificial diffusion due to some appropriate FEM stabilization as described before. The solution of nonlinear algebraic systems like (4.11) is a rather difficult task and many aspects, namely the treatment of the nonlinearity and of the incompressibility as well as the outer control of the couplings, need to be taken into account. Consequently, this leads to a great variety of incompressible flow solvers which are closely related to one another but exhibit considerable differences in terms of their stability, convergence, and efficiency. The Multilevel Pressure Schur Complement (MPSC) approach outlined below makes it possible to put many existing solution techniques into a common framework and to combine their advantages so as to obtain better run-time characteristics.

The fully discretized Navier-Stokes equations (4.11) as well as the linear subproblems to be solved within the outer iteration loop for a fixed-point defect correction or, with a similar structure, for a Newton-like method admit the following representation:

$$\begin{bmatrix} A & \Delta t B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p_\rho \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix}. \tag{4.15}$$

In general, we have $A = M + \beta N(\mathbf{u})$, with $\beta = -\theta \Delta t$ for time-dependent problems. If the operator A is nonsingular, the velocity can be formally expressed as

$$\mathbf{u} = A^{-1}(\mathbf{g} - \Delta t B p_\rho) \tag{4.16}$$

and plugged into the discretized continuity equation

$$B^T \mathbf{u} = 0 \tag{4.17}$$

which gives a scalar *Schur complement* equation for the pressure only

$$B^T A^{-1} B p_\rho = \frac{1}{\Delta t} B^T A^{-1} \mathbf{g}. \tag{4.18}$$

Thus, the coupled system (4.15) can be handled as follows:

1. Solve the Pressure Schur Complement (PSC) equation (4.18) for p_ρ.
2. Substitute p_ρ into the relation (4.16) and compute the velocity $\mathbf{u}$.

It is worth mentioning that the matrix A^{-1} is full and should not be assembled explicitly. Instead, an auxiliary problem is to be solved by a direct method or by inner iterations. For instance, the velocity update (4.16) is equivalent to the solution of the discretized momentum equation $A\mathbf{u} = \mathbf{g} - \Delta t B p_\rho$. Likewise, the matrix $S := B^T A^{-1} B$ is never generated in practice. Doing so would be prohibitively expensive in terms of CPU time and memory requirements. It is instructive to consider

a preconditioned Richardson method which yields the following **basic iteration** for the PSC equation:

$$p_\rho^{(l+1)} = p_\rho^{(l)} - C^{-1}\left[S p_\rho^{(l)} - \frac{1}{\Delta t} B^T A^{-1} \mathbf{g} \right], \quad l = 0, \ldots, L-1. \tag{4.19}$$

Here, C has to be chosen as a suitable preconditioner to S but being easier to 'invert' in an iterative way. The number of PSC cycles L can be fixed or chosen adaptively so as to achieve a prescribed tolerance for the residual. The basic idea behind the family of global MPSC schemes is the construction of globally defined additive preconditioners for the Schur complement operator $S = B^T A^{-1} B$. Recall that the matrix A has the structure

$$A := M + \beta K(\mathbf{u}) + \gamma L, \tag{4.20}$$

where $\beta = -\theta \Delta t$ and $\gamma = \nu\beta$. Unfortunately, even today it is still a very challenging task to construct a matrix $\tilde{A}$ and a preconditioner $C = B^T \tilde{A}^{-1} B$ that would be a sufficiently good approximation to all three components of A and S, respectively; particularly for the convective part with $K(\mathbf{u})$. Therefore, one may start with developing individual preconditioners for the reactive (M) and diffusive (L) part, while the convective (K) part is neglected by applying this special kind of operator splitting. In our case, the Reynolds numbers in the considered flow configurations are so far quite small, so that this approach can be justified, particularly if small time steps are used to resolve the complex dynamical behaviour. Therefore, the (lumped) mass matrix M proves to be a reasonable approximation to the complete operator A, so that our basic iteration (4.19) for the pressure Schur complement equation

$$p_\rho^{(l+1)} = p_\rho^{(l)} + \left[B^T M^{-1} B \right]^{-1} \frac{1}{\Delta t} B^T A^{-1} \left[\mathbf{g} - \Delta t B p_\rho^{(l)} \right] \tag{4.21}$$

can be interpreted and implemented as a *discrete projection scheme*, if $L = 1$, such as those proposed in [12, 22]. Here, the important step is that for the chosen Stokes element pair, Q_2/P_1, the matrix $P := B^T M^{-1} B$ can be explicitly built up relatively easily even in a domain decomposition framework due to the chosen discontinuous pressure. Then, the main algorithmic steps are as follows [60]:

Step 1. Solve the 'viscous Burgers' equation for $\tilde{\mathbf{u}}$

$$A\tilde{\mathbf{u}} = \mathbf{g} - \Delta t B p_\rho^{(l)}.$$

Step 2. Solve the discrete 'Pressure-Poisson' problem

$$P q_\rho = \frac{1}{\Delta t} B^T \tilde{\mathbf{u}}.$$

Step 3. Correct the pressure and the velocity

$$p_\rho^{(l+1)} = p_\rho^{(l)} + q_\rho, \qquad \mathbf{u} = \tilde{\mathbf{u}} - \Delta t M^{-1} B q_\rho.$$

In essence, the right-hand side of the momentum equation is assembled using the old pressure iterate, and the intermediate velocity $\tilde{\mathbf{u}}$ is projected onto the subspace of solenoidal functions so as to satisfy the constraint $B^T\mathbf{u} = 0$. Moreover, the matrix P corresponds to a mixed discretization of the Laplacian operator [22] so that this method is a discrete analogue of the classical projection schemes derived by Chorin ($p_\rho^{(0)} = 0$) and Van Kan ($p_\rho^{(0)} = p_\rho(t_n)$) via operator splitting for the continuous problem.

Next, we apply this special operator-splitting approach to the full multiphase flow system with a discontinuous density $\rho(\phi)$ and viscosity $\mu(\phi)$ distribution, that means

$$\rho(\phi)\left[\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right] - \nabla\cdot\left(\mu(\phi)\left[\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right]\right) + \nabla p = \rho(\phi)\mathbf{g} + \mathbf{f}_{\Gamma,\sigma}(\phi), \tag{4.22}$$

$$\frac{\partial \phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = 0, \qquad \nabla\cdot\mathbf{u} = 0. \tag{4.23}$$

After discretization in space and time, we obtain again a system of nonlinear algebraic equations which can be written in a matrix form as follows:

$$A_u\left(\mathbf{u}^{n+1}, \phi^{n+1}\right)\mathbf{u}^{n+1} + \Delta t\, F\left(\phi^{n+1}\right) + \Delta t\, B p^{n+1} = \mathbf{g}_u, \tag{4.24}$$

$$A_\phi\left(\mathbf{u}^{n+1}\right)\phi^{n+1} = g_\phi, \qquad B^T\mathbf{u}^{n+1} = 0. \tag{4.25}$$

Note that Eq. (4.24) in contrast to (4.11) and (4.14) is multiplied with $\rho(\phi)$, which gives rise to the modified operators M_ρ, $K_\rho(\mathbf{u})$, and L_μ. Here and below the superscript $n+1$ refers to the time level, while subscripts identify the origin of discrete operators (u for the momentum equation and ϕ for the Level Set equation); moreover, ρ and μ are evaluated w.r.t. the old time level t^n which makes this formulation semi-implicit. Note that we have the freedom of using different finite element approximations and discretization schemes for the velocity $\mathbf{u}$ and the indicator function ϕ, and the discrete problem (4.24)–(4.25) can be solved again in the framework of the discrete projection method. For relatively small time steps, this strategy works very well, and simulation software can be developed in a modular way making use of optimized multigrid solvers. Consequently, in the simplest case (just one outer iteration per time step), the sequence of algorithmic steps to be performed is as follows:

Step 1. Compute $\tilde{\mathbf{u}}$ from the momentum equation

$$A_u\left(\tilde{\mathbf{u}}, \phi^n\right)\tilde{\mathbf{u}} = \mathbf{g}_u - \Delta t\, F\left(\phi^n\right) - \Delta t\, B p^n.$$

Step 2. Solve the discrete Pressure-Poisson problem

$$P_\rho q = \frac{1}{\Delta t} B^T\tilde{\mathbf{u}} \quad \text{with } P_\rho := B^T M_\rho^{-1} B.$$

Step 3. Correct the pressure and the velocity

$$p^{n+1} = p^n + q, \qquad \mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \Delta t M_\rho^{-1} B q.$$

Step 4. Solve the Level Set equation for ϕ

$$A_\phi\big(\mathbf{u}^{n+1}\big)\phi^{n+1} = g_\phi.$$

Due to the nonlinearity of the discretized convective terms, resp., of the reinitialisation step, iterative defect correction or Newton-like methods, resp., corrections via redistancing, must be invoked in Steps 1 and 4. However, due to the assumed relatively small time steps, such nonlinear iteration methods are not critical for the complete flow simulation.

4.4 The FEM-Level Set-dG(1) Approach

Our chosen Level Set approach is based on a first-order Discontinuous Galerkin discretization in space, dG(1)-FEM, that means on piecewise linear polynomials. In the following, we will discuss the corresponding techniques for the discretization of the advection equation, for the treatment of the surface tension force, and for the reinitialisation procedure.

4.4.1 Discontinuous Galerkin Upwinding for the Level Set Approach

There are several ways to approximate and solve Discontinuous Galerkin approximations for the Level Set function ϕ [11, 15, 36, 41]. The general form of the Level Set transport equation involving the normal front velocity can for instance be solved directly by using a Runge–Kutta dG-formulation for the Hamilton–Jacobi equations [27, 35]. The starting point to introduce our discretization of the Level Set transport equation is

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \tag{4.26}$$

with a given velocity field $\mathbf{u}$. In our case $\mathbf{u}$ is taken as the convective velocity from the Navier–Stokes solver and must accordingly be updated in each time step. We have $\mathbf{u} \cdot \mathbf{n} = u_n$, where $\mathbf{n}$ is the unit normal to the interface Γ according to (4.7). The Level Set equation (4.26) can thus be rewritten as

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = \phi \nabla \cdot \mathbf{u}. \tag{4.27}$$

The reformulated Level Set equation above is simply a linear convection or advection equation in conservative formulation with a source term on the right hand

side. We continue to rewriting it in weak form by introducing a triangulation, $\mathscr{M}_h$, of the domain Ω where $\mathscr{E}$ is an element $\mathscr{E} \in \mathscr{M}^h$. We are thus seeking an approximated solution in the following space

$$V_h = \left\{ v_h \in L^\infty(\Omega) : v_h|_{\mathscr{E}} \in V_h(\mathscr{E}), \forall \mathscr{E} \in \mathscr{M}^h \right\}.$$

Here, $V_h(\mathscr{E})$ denotes the local discrete test and trial spaces. The corresponding derivation follows by multiplying the equation (4.26) by a suitably chosen test function after which partial integration over each element $\mathscr{E}$ is performed. If the trial solution space is accordingly discretized as $\phi_h \in V_h(\mathscr{E})$, this results in

$$\int_{\mathscr{E}} v_h \frac{\partial \phi_h}{\partial t} dx = \int_{\mathscr{E}} \phi_h \mathbf{u} \cdot \nabla v_h dx - \int_{\partial \mathscr{E}} v_h \phi_h \mathbf{u} \cdot \mathbf{n}_{\mathscr{E}} ds + \int_{\mathscr{E}} v_h \phi_h \nabla \cdot \mathbf{u} dx,$$
$$\forall v_h \in V_h(\mathscr{E}), \tag{4.28}$$

where $\mathbf{n}_{\mathscr{E}}$ is the outward pointing unit normal belonging to the element $\mathscr{E}$. The fluxes on the internal boundaries are twofold defined since the underlying test and trial spaces are discontinuous. This is handled by replacing the outer flux in the last term of the right-hand side of Eq. (4.28) with a numerically upwinded flux, that is

$$\int_{\mathscr{E}} v_h \frac{\partial \phi_h}{\partial t} dx = \int_{\mathscr{E}} \phi_h \nabla \cdot (\mathbf{u} v_h) dx - \int_{\partial \mathscr{E}} v_h \phi_h^{up} \mathbf{u} \cdot \mathbf{n}_{\mathscr{E}} ds, \quad \forall v_h \in V_h(\mathscr{E}). \tag{4.29}$$

The upwinding flux is calculated as

$$\phi^{up} = \begin{cases} \phi^-, & \text{if } \mathbf{u} \cdot \mathbf{n}_{\mathscr{E}} \geq 0, \\ \phi^+, & \text{otherwise,} \end{cases}$$

where ϕ^- and ϕ^+ are defined as

$$\phi^- = \lim_{\epsilon \to 0^-} \phi(\mathbf{x} + \epsilon \mathbf{n}_\epsilon, t),$$
$$\phi^+ = \lim_{\epsilon \to 0^+} \phi(\mathbf{x} + \epsilon \mathbf{n}_\epsilon, t).$$

In other words this means that ϕ^{up} is the value of ϕ taken from an upwind element at an element interface.

In our approach, Eq. (4.29) is discretized in space by firstly constructing the triangulation $\mathscr{M}_h$ by subdivision in the hexahedral elements $\mathscr{E}$. Furthermore, both the test and trial function spaces, v_h and ϕ_h, are constructed by employing linear first-order polynomial basis functions on each element $\mathscr{E}$, the so-called dG(1) approach. These basis functions are completely determined by interior nodes of the element and are thus discontinuous at inter-element edges. Moreover, the discretization in time utilizes as before the standard second-order Crank–Nicolson scheme as described for instance in [61].

4.4.2 Treatment of Surface Tension Effects

Surface tension effects are taken into account through the following force balance at the interface Γ:

$$[\mathbf{u}]|_\Gamma = 0, \qquad \left[-p\mathbf{I} + \mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right]\Big|_\Gamma \cdot \mathbf{n} = \sigma\kappa\mathbf{n}.$$

Here $\mathbf{n}$ is the unit normal at the interface pointing into Ω_1, $[\mathsf{A}]|_\Gamma = \mathsf{A}|_{\Omega_1\cap\Gamma} - \mathsf{A}|_{\Omega_2\cap\Gamma}$ denotes the jump of a quantity A across the interface, σ is the surface tension coefficient, and κ is the curvature of the interface Γ. The first condition implies continuity of the velocity across the interface, whereas the second describes the force balance on Γ. Two strategies are often used to handle the curvature term, either to rewrite it as a volume force, that means

$$\mathbf{f}_{st} = \sigma\kappa\mathbf{n}\delta(\Gamma, \mathbf{x}),$$

where $\delta(\Gamma, \mathbf{x})$ is the Dirac delta function localizing the surface tension forces to the interface, or to introduce the Laplace–Beltrami operator Δ_Γ on the interface, that means

$$\kappa\mathbf{n} = \Delta_\Gamma \,\mathrm{id}$$

and integrating the corresponding term in the weak formulation of the problem by parts [1, 17]. In the case of our current explicit treatment we get

$$(\mathbf{f}_{st}, \mathbf{v}) = \int_{\Gamma^n} \sigma\kappa^n\mathbf{n}^n \cdot \mathbf{v}\, d\Gamma, \tag{4.30}$$

where the superscript n denotes the previous time level. The extension of the surface integrals into volumetric ones can be obtained by the indicated incorporation of the Dirac Delta function $\delta = \delta(\Gamma, \mathbf{x})$, which has the value ∞ at the location of the interface, $\phi = 0$, and zero elsewhere, that means

$$(\mathbf{f}_{st}, \mathbf{v}) = \int_\Omega \sigma\kappa^n\mathbf{n}^n \cdot \mathbf{v}\delta\left(\Gamma^n\right) dx. \tag{4.31}$$

According to the applied CSF approach we approximate the Dirac Delta function δ by a continuous regularized one, which is a smooth function in the vicinity ϵ of the interface:

$$\delta(\phi) = \begin{cases} \phi < 0, & \max(0, \frac{1}{\epsilon} + \frac{1}{\epsilon^2}\phi), \\ \phi \geq 0, & \max(0, \frac{1}{\epsilon} - \frac{1}{\epsilon^2}\phi). \end{cases} \tag{4.32}$$

Since the interface normal $\mathbf{n}^n$ and curvature κ^n are higher order derivatives of the Level Set function ϕ^n, their distributions can be obtained by a combination of appropriate projection and gradient recovery techniques. Accordingly, the continuous (piecewise trilinear) interface normal $\mathbf{n}^n_{Q_1}$ is obtained by L_2-projection (and normalization) from the piecewise discontinuous P_1 space into the continuous Q_1 space.

Finally, the continuous approximation $\kappa^n_{Q_1}$ of the curvature κ^n is reconstructed via L_2-projection, too,

$$\int_\Omega \kappa^n_{Q_1} w \, dx = -\int_\Omega w \, \nabla \cdot \mathbf{n}^n_{Q_1} \, dx, \tag{4.33}$$

where w denotes the test functions from the conforming trilinear Q_1 space.

One of the remaining challenging problems is the *capillary time step restriction* which couples the time step size with the (local) mesh size h and $1/\sigma$ leading to very high computational cost due to such strict stability constraints. Beside the classical work by Bänsch, who developed a semi-implicit approach for front tracking, the FEM-Level Set approach by Hysing [28] is one of the very few attempts for interface capturing methods, which is in the focus of our future research on 3D multiphase flow problems. Very recently, an alternative method containing a survey on this problem and existing solution strategies was published by Sussmann [58]. However, it still has to be stated that the combination of adaptive Level Set or VOF methods on locally adapted meshes shows severe numerical problems if configurations with large surface tension shall be simulated in an accurate, robust, and efficient way. Moreover, the challenges further increase for non-Newtonian multiphase fluids, for instance for Power Law models ('shear thinning' [13]) or for viscoelastic fluids [68] which even for single-phase flows lead to huge problems for large Weissenberg numbers. Nevertheless, we are convinced that the described FEM-Level Set techniques have the potential to solve these challenging problems in future.

As a final comment, in the framework of variational formulations, the corresponding volume integral can be reduced to a boundary integral which serves as a natural boundary condition at the free interface [48, 56]. Moreover, if partial integration of the *Laplace-Beltrami* operator is applied in tangential direction of the interface [1, 2, 14, 18, 24, 37] then the calculation of the second derivatives of ϕ for the curvature can be omitted which can be used for very efficient evaluations of the surface tension force in combination with Level Set functions satisfying the distance property. This is in contrast to the usual finite difference approaches which require a less accurate *Continuum Surface Force* (CSF) approximation of the (singular) Delta function [6]. The above-mentioned alternative treatment of the surface tension force term is in the scope of our forthcoming studies.

4.4.3 Reinitialization Procedure for LS-dG(1)

For the accurate calculation of the normal vector and curvature, as defined in (4.7), and hence for the accurate position and shape of the dynamic interface, one has to take care that ϕ satisfies—at least near the interface Γ—the distance property which typically is achieved via appropriate postprocessing of a given numerical approximation $\tilde{\phi}$. Since the direct *reinitialisation* $\phi_i := \text{sign}(\tilde{\phi}_i)\,\text{dist}(\mathbf{x}_i, \Gamma)$ is very expensive, one way to do the corresponding corrections is to solve the so-called Eikonal equation $|\nabla\phi| = 1$ [30, 33] with boundary conditions $\phi = 0$ on $\Gamma = \{\mathbf{x} \in$

$\Omega \mid \tilde{\phi}(\mathbf{x}) = 0\}$. Typical methods are based on *fast marching* [53] or *fast sweeping* [69], while another approach is based on pseudo-timestepping for this nonlinear equation which leads to a Hamilton–Jacobi PDE:

$$\frac{\partial \phi}{\partial \tau} = \operatorname{sign}(\tilde{\phi})\big(1 - |\nabla \phi|\big), \quad \phi|_{\tau=0} = \tilde{\phi}. \tag{4.34}$$

Corresponding numerical approaches exploit that this problem can be written as a (nonlinear) transport equation

$$\frac{\partial \phi}{\partial \tau} + \mathbf{w} \cdot \nabla \phi = \operatorname{sign}(\tilde{\phi}), \quad \text{with } \mathbf{w} = \operatorname{sign}(\tilde{\phi}) \frac{\nabla \phi}{|\nabla \phi|}. \tag{4.35}$$

By stability reasons, the (discontinuous) sign function is typically replaced by a smoothed approximation which may lead to loss of accuracy and shift of the free interface. In the framework of FEM, the *interface local projection* of Parolini [47] helps, particularly for piecewise linear functions leading to a constant gradient vector, which combines the advantages of direct and PDE-based reinitialisation. Then, the correction of $\tilde{\phi}$ mostly consists of three steps:

1. In mesh cells which contain the free boundary Γ, an exact reconstruction via (piecewise constant) gradient is applied.
2. Use a L_2 projection to obtain the best approximation of ϕ near Γ.
3. Outside of the 'surface domain' Ω_{int}, solve the equation (4.35) using the already calculated values of ϕ at the boundary of Ω_{int} as Dirichlet boundary conditions.

According to our implementation, the reinitialization of the Level Set distribution is based on the advantages offered by the Discontinuous Galerkin Finite Element Method dG(1). This particularly means that we perform segregated reinitialization procedures on different groups of elements. The identified groups are as follows:

- Elements intersected by the interface, we denote them by $\mathscr{E} \subset \mathscr{M}^0$.
- A few layers of elements in the positive direction ($\phi > 0$) from the interface, $\mathscr{E} \subset \mathscr{M}^+$.
- A few layers of elements in the negative direction ($\phi < 0$) from the interface, $\mathscr{E} \subset \mathscr{M}^-$.
- The rest of the domain, these are the elements $\mathscr{E} \subset \mathscr{M}^\infty$.

Such a segregated approach enables us to get rid of the discontinuity that the sign function $S(\phi)$ exhibits at elements intersected by the interface. Moreover, it reduces the computational overhead since the PDEs are computed in a reduced computational domain only. Summarizing, the developed algorithm for the reinitialization is as follows:

Step 1. Direct reinitialization for $\mathscr{E} \subset \mathscr{M}^0$:

$$\phi^n \overset{|\nabla \phi|=1}{\longrightarrow} \phi^{n+1}.$$

Step 2. PDE-based solution for $\mathscr{E} \subset \mathscr{M}^+$ with

$$\frac{\partial \phi}{\partial \tau} + \mathbf{n} \cdot \nabla \phi = +1.$$

Step 3. PDE-based solution for $\mathscr{E} \subset \mathscr{M}^-$ with

$$\frac{\partial \phi}{\partial \tau} - \mathbf{n} \cdot \nabla \phi = -1.$$

Step 4. Prescription of far field values for $\mathscr{E} \subset \mathscr{M}^\infty$: $\phi_{\mathrm{RI}}^{n+1} = \phi^\infty$.

Here

$$\mathbf{n} := \mathbf{n}^n = \frac{\nabla \phi^n}{|\nabla \phi^n|}.$$

The coupling between the individual groups of elements is achieved by imposing of boundary conditions from $\mathscr{E} \subset \mathscr{M}^0$ for the PDE-based reinitialization which is treated via the Fictitious Boundary Method approach [39]. One has to keep in mind that the discontinuous sign function does not cause a problem in Steps 2 and 3 since the discontinuity has been treated already in Step 1. Additionally, the Level Set function can be corrected due to mass loss which typically is performed by adding an appropriate constant c_ϕ so that the total volume of both phases remains constant [56]. Moreover, further improvements can be obtained via high-order discretization and grid adaptivity [11] which is a subject of ongoing research.

4.5 Numerical Simulations

This section contains several numerical studies for validating and evaluating the methodology described in the previous sections.

4.5.1 Single-Phase Flow Around a Cylinder

The first incompressible flow problem to be dealt with, particularly to demonstrate the accuracy of the high-order Q_2/P_1 approach, is the well-known benchmark *Flow around cylinder* developed in 1995 for the priority research program "Flow simulation on high-performance computers" under the auspices of DFG, the German Research Association [64]. This project was intended to facilitate the evaluation of various numerical algorithms for the incompressible Navier-Stokes equations in the laminar flow regime. A quantitative comparison of simulation results is possible on the basis of relevant flow characteristics such as pressure values as well as drag and lift coefficients, for which sufficiently accurate reference values are available (see also: www.featflow.de/en/benchmarks/ff_benchmarks.html).

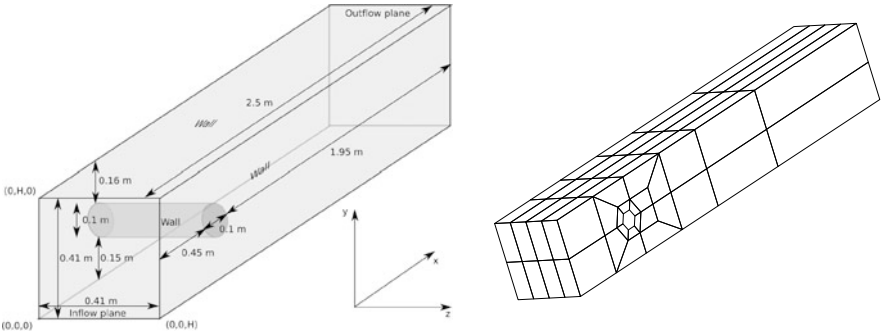


Fig. 4.2 Geometry and a coarse mesh for the 'Flow around cylinder' benchmark

Table 4.1 Mesh convergence results (levels 2 to 6) in terms of drag, lift, and pressure difference for the 'DFG Flow around cylinder problem' at $Re = 20$. Comparison of our results with reference results [5, 32]. ΔP refers to the pressure difference (front/back) on the cylinder and C_D and C_L are the normalized ($\frac{1}{2}\rho U_{\text{mean}}^2 L_{\text{cyl}} D_{\text{cyl}}$) drag and lift coefficients

Level	ΔP	C_D	C_L	NEL	NDOF($\mathbf{u}$, p)
2	0.171956	6.01954	0.012316	768	21,560
3	0.171553	6.13973	0.009569	6,144	199,200
4	0.171156	6.17433	0.009381	49,152	1,482,816
5	0.171031	6.18261	0.009387	393,216	11,432,064
6	0.171022	6.18465	0.009397	3,145,728	89,760,016
Authors	Reference values				
Braack [5]	0.171007	6.18533	0.009401	1,000,000	40,000,000
John [32]	0.170779	6.18533	0.009401	2,000,000	55,000,000

Here, we consider the steady incompressible flow around a cylinder with circular cross-section (see Fig. 4.2). An in-depth description of the geometrical details and boundary conditions can be found in [4, 64] which contain all relevant information regarding this benchmark configuration. The flow at $Re = 20$ is actually dominated by diffusion and could be simulated by the standard Galerkin method without any extra stabilization. The corresponding results are shown in Table 4.1 and demonstrate the high quality of the Q_2/P_1 approach compared to quasi-reference values from the literature [5, 32].

4.5.2 Two-Phase Flow of a Rising Bubble

The rising bubble configurations described in this section were chosen as the ones established by the numerical studies of van Sint Annaland et al. [54] in order to

Table 4.2 Resulting the Reynolds numbers obtained for the different configurations. The subscripts E and S stand for empirical and simulational reference values from Grace [8] and van Sint Annaland [54], respectively. The last four values refer to our simulation results obtained on the meshes A and B and on the refinement levels 2, 3, and 4

Case	Shape	Mo	Eo	Re_E	Re_S	Re_{mAl2}	Re_{mAl3}	Re_{mBl3}	Re_{mBl4}
B	Ellipsoidal	0.100	9.71	4.6	4.3	5.50	5.50	5.60	5.60
C	Skirted	0.971	97.1	20.0	18.0	17.7	18.0	18.0	18.0
D	Dimpled	1000	97.1	1.5	1.7	2.00	2.03	2.03	2.03

validate the implementation of our Level Set approach. According to the mentioned studies, the cases B, C, and D were analysed which results in a considerable deformation of the initial bubble. The ratios of physical properties ($\rho_g : \rho_l$ and $\mu_g : \mu_l$) of the present phases were set to (1 : 100). The ratios of the bubble diameter, d_b, with respect to the domain sizes, a_x, a_y, a_z, were $(d_b : a_x : a_y : a_z) = (3 : 10 : 10 : 20)$. The values of the interfacial tension coefficient σ_{gl} and gravitational acceleration g_z for the simulations were set based on the characteristic Eötvös and Morton numbers defined as in [8]:

$$\mathrm{Mo} = \frac{g_z \mu_l^4 \Delta\rho_{gl}}{\rho_l^2 \sigma_{gl}^3}, \qquad \mathrm{Eo} = \frac{g_z \Delta\rho_{gl} d_b^2}{\sigma_{gl}}. \tag{4.36}$$

As a result of the given settings the bubbles deform to a final shape and they reach an equilibrium rising velocity, v_∞, characterized by the Reynolds number defined as in [8]:

$$Re = \frac{\rho_l v_\infty d_b}{\mu_l}. \tag{4.37}$$

Since the Level Set approach by its nature does not preserve the mass of the individual phases, certain mass correction techniques were incorporated to prevent artificial '*mass transformation*' from one phase to another. To this end we adopted a simple but efficient method proposed by Smolianski [56] which elevates the level set function at every time step with a limited constant $\min(d_\varepsilon, \max(-d_\varepsilon, c_\phi))$, where d_ε is related to the characteristic element size and c_ϕ is a value enforcing absolute mass conservation. According to our experience setting d_ε to 3 % of the characteristic element size already prevents the occurrence of permanent mass loss.

In order to achieve mesh-independent simulation results in terms of bubble shape and terminal rising velocity, we performed the simulations on two sets of meshes of two consequent levels of refinements (mesh A with refinement level 2, 3 and mesh B refinement level 3, 4). As it can be seen from Fig. 4.3, which displays the equilibrium bubble shapes centered with respect to their center of mass, the bubble shapes converge fairly well with increasing mesh resolution, especially in cases B and D. The terminal rise velocities compared with the empirical predictions of Grace [8] and numerical predictions of van Sint Annaland [54] are given in Table 4.2. Despite the mesh-independent properties of the obtained results, the comparison of the

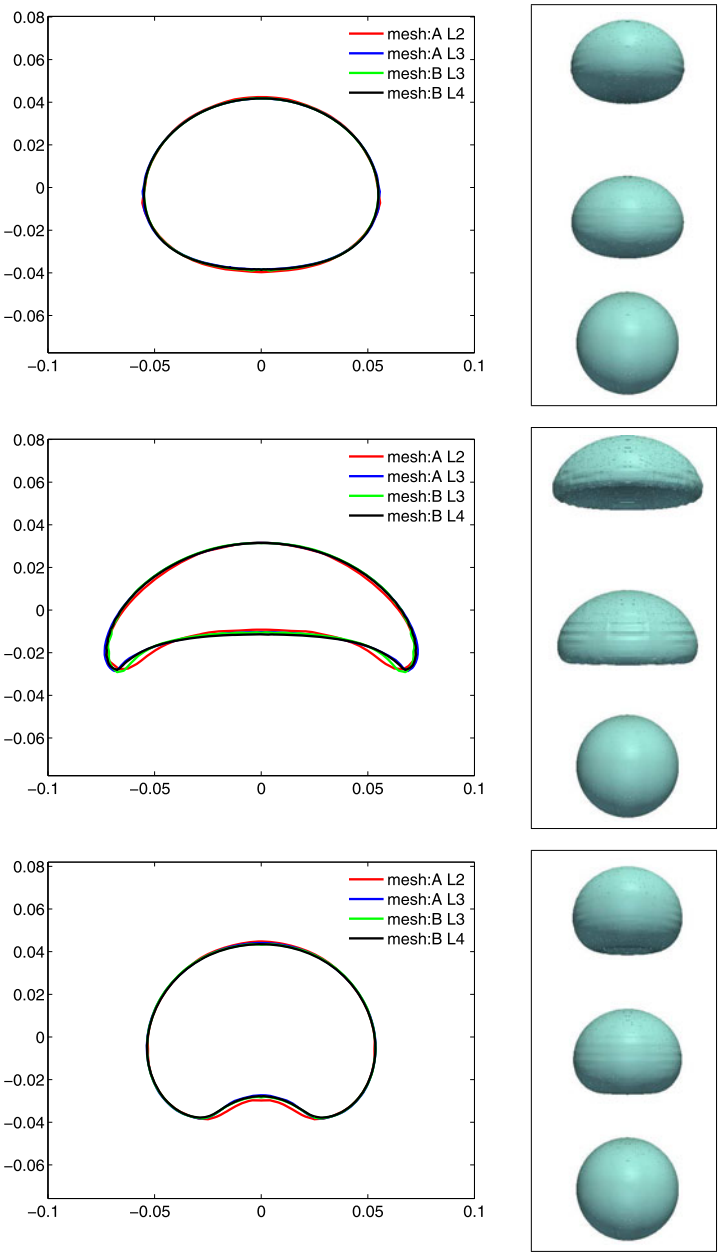


Fig. 4.3 *Left*: Cutplanes of continuous reconstructions of the interphase for the equilibrium bubble shapes. *Right*: Time evolution of the bubble shapes (from bottom to top). The cases are organized as: *Top*—case B—Eo = 9.71, Mo = 0.1; *Middle*—case C—Eo = 97.1, Mo = 0.971; *Bottom*—case D—Eo = 97.1, Mo = 1000

terminal rise velocities shows a weaker correlation of our results with the empirical predictions of Grace than it was the case in van Sint Annaland's computational studies. This contradiction leaves behind the challenges for further numerical analysis, possibly leading to a benchmark problem to which other researchers will also be welcome to contribute, as was the case with the well-known 2D rising bubble problem [31].

4.5.3 Droplet Dripping Simulation

The corresponding experimental setup involves a two-phase problem consisting of a glucose-water mixture (as a continuous phase) and silicon oil (as a dispersed phase). The measurements are restricted to the so-called dripping mode. This mode is characterized by relatively low volumetric flow rates and by the fact that the droplets are generated in the near vicinity of the capillary so that the stream length is comparable with the size of the generated droplets. Since the temperature is kept at a constant value during the whole experiment, all physical properties of the present phases are constant. The experimental measurements were realized (by the group of Prof. Walzel, BCI, TU Dortmund) to obtain statistically averaged quantities such as droplet size, droplet generation frequency and stream length. These experimentally measured quantities are compared with our subsequent simulation results.

The basic units used to define the derived quantities are the following ones:

$$[\text{length}] = \text{dm}, \qquad [\text{time}] = \text{s}, \qquad [\text{mass}] = \text{kg}.$$

The list of physical quantities is as follows (Fig. 4.4):

$$g_z = -9.81\ \text{m}\,\text{s}^{-2} = -98.1\ \text{dm}\,\text{s}^{-2},$$
$$\sigma = 0.034\ \text{N}\,\text{m}^{-1} = 0.034\ \text{kg}\,\text{s}^{-2},$$
$$\rho_C = 1340\ \text{kg}\,\text{m}^{-3} = 1.34\ \text{kg}\,\text{dm}^{-3},$$
$$\rho_D = 970\ \text{kg}\,\text{m}^{-3} = 0.97\ \text{kg}\,\text{dm}^{-3},$$
$$\mu = \mu_C = \mu_D = 500\ \text{mPa}\,\text{s} = 0.050\ \text{kg}\,\text{dm}\,\text{s}^{-1}.$$

The list of geometrical parameters reads:

$[\text{domain size}] = [-0.15 : 0.15] \times [-0.15 : 0.15] \times [0.0 : 1.2]\ \text{dm}^3$,
$[\text{inner capillary radius}] = R_1 = 0.015$ dm,
$[\text{outer capillary radius}] = R_2 = 0.030$ dm,
$[\text{primary phase inflow radius}] = R_3 = 0.15$ dm.

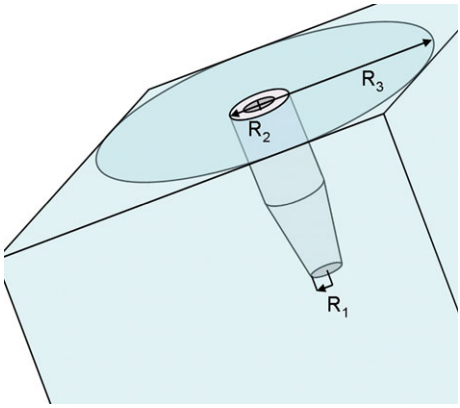


Fig. 4.4 A sketch of the benchmark domain

The boundary conditions imposed on the inflow velocity are the following:

$$w = \begin{cases} a_2(R_1 - r)(R_1 + r) & \text{if } 0 < r < R_1 \text{ (a dispersed phase)}, \\ a_1(R_3 - r)(r - R_2) & \text{if } R_2 < r < R_3 \text{ (a continuous phase)}, \\ 0 & \text{otherwise.} \end{cases}$$

The parameters a_1 and a_2 are defined to achieve the required volumetric flow rates:

$$\dot{V}_C = \int_{R_2}^{R_3} \left(2\pi r a_1 (R_3 - r)(r - R_2)\right) dr$$

$$= -2\pi a_1 \left[\frac{r^4}{4} - (R_2 + R_3)\frac{r^3}{3} + R_2 R_3 \frac{r^2}{2}\right]_{R_2}^{R_3} = \frac{\pi a_1}{6}(R_2 + R_3)(R_3 - R_2)^3.$$

$$\dot{V}_D = \int_0^{R_1} \left(2\pi r a_2 (R_1 - r)(R_1 + r)\right) dr = 2\pi a_2 \left[\frac{R_1^2 r^2}{2} - \frac{r^4}{4}\right]_0^{R_1} = \frac{\pi a_2}{2} R_1^4.$$

The volumetric flow rates for the simulations are set to:

$$\dot{V}_C = 99.04 \text{ ml min}^{-1} = 99.04 \text{ cm}^3 \text{ min}^{-1} = 99.04 \frac{10^{-3} \text{ dm}^3}{60 \text{ s}}$$

$$= 1.65 \times 10^{-3} \text{ dm}^3 \text{ s}^{-1},$$

$$\dot{V}_D = 3.64 \text{ ml min}^{-1} = 3.64 \text{ cm}^3 \text{ min}^{-1} = 3.64 \frac{10^{-3} \text{ dm}^3}{60 \text{ s}}$$

$$= 6.07 \times 10^{-5} \text{ dm}^3 \text{ s}^{-1},$$

which is guaranteed by setting $a_1 = 10.14 \text{ dm}^{-1} \text{ s}^{-1}$ and $a_2 = 763.7 \text{ dm}^{-1} \text{ s}^{-1}$.

The resulting process leads to a pseudo-steady state, where the droplet separation happens according to the so-called dripping mode. The frequency of the given mode is $f = 0.60 \text{ Hz}$ (cca $0.58 \text{ Hz}^{\text{exp}}$), which produces droplets of size $d = 0.058 \text{ dm}$ (cca

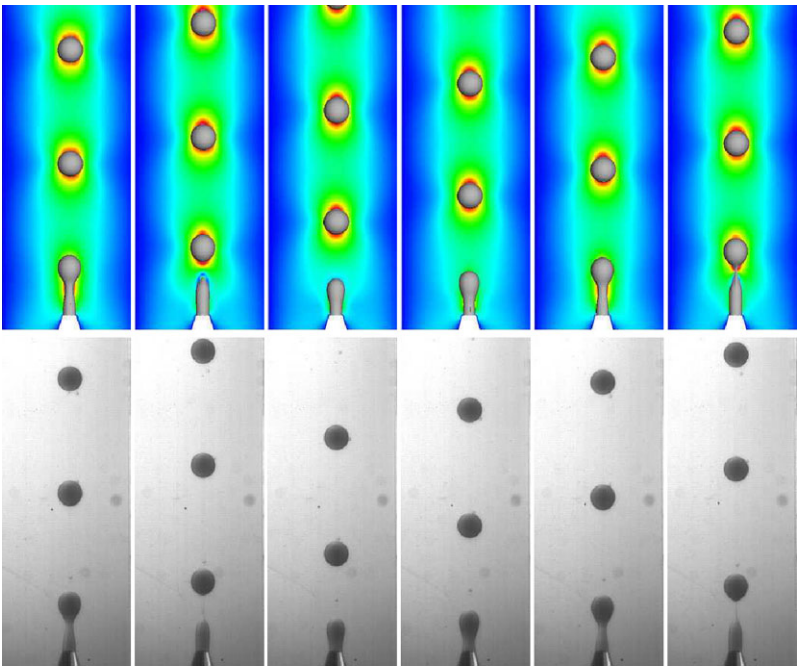


Fig. 4.5 A sequence of one droplet separation compared with experimental measurements

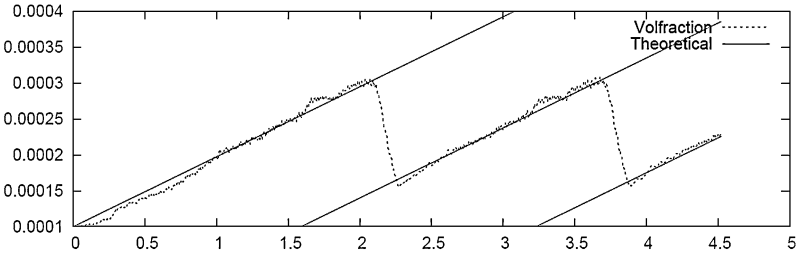


Fig. 4.6 Evolution of the volume of the secondary phase. Theoretical lines are characterized by the slope $q = 6.07 \times 10^{-5}\ \mathrm{dm^3\,s^{-1}}$

$0.062\,\mathrm{dm}^{\mathrm{exp}}$). The maximum stream length during the process is $L = 0.102\,\mathrm{dm}$ (cca $0.122\,\mathrm{dm}^{\mathrm{exp}}$). The snapshots of one full droplet generation compared with experimental measurements are given in Fig. 4.5. The time evolution of the volume of the secondary phase is given in Fig. 4.6. As it can be seen, the increase of the volume of the dispersed phase follows the theoretically expected trend in a reasonable way despite the fact that the mass correction technique (previously described in Sect. 4.5.2) was not activated.

4.6 Summary

In this contribution, we have shown that the realization of a new FEM-Level Set approach in the framework of Discontinuous Galerkin Finite Elements together with special PDE-based reinitialization techniques leads to very efficient simulation tools for modelling multiphase flow problems. The implemented parallel 3D multiphase flow solver has been validated in the case of the rising bubble and for the droplet dripping problem. A detailed description of these problems together with the obtained results—which are accurate and in fairly good agreement with the corresponding empirical data—are left in the form of a benchmark proposal for the engineering community.

Acknowledgements The authors would like to thank the German Research Foundation (DFG) for partially supporting the work under the grants Sonderforschungsbereich SFB708 (TP B7) and SPP 1423 (Tu102/32-1) and the group of Prof. Walzel at TU Dortmund for the experimental measurements supported by the grants Paketantrag PAK178 (Tu102/27-1, Ku1530/5-1).

References

1. Bänsch E (1998) Numerical methods for the instationary Navier-Stokes equations with a free capillary surface. Habil. thesis, Universität Freiburg, Freiburg
2. Bänsch E (2001) Finite element discretization of the Navier-Stokes equations with a free capillary surface. Numer Math 88(2):203–235. doi:10.1007/s002110000225
3. Barth TJ, Sethian JA (1998) Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains. J Comput Phys 145(1):1–40
4. Bayraktar E, Mierka O, Turek S (2011) Benchmark computations of 3D laminar flow around a cylinder with CFX, OpenFOAM and FEATFLOW. Internat J Comput Sci Engrg. To appear. Also available as: Ergebnisberichte des Instituts für Angewandte Mathematik Nummer 433, Fakultät für Mathematik, TU, Dortmund
5. Braack M, Richter T (2006) Solutions of 3D Navier-Stokes benchmark problems with adaptive finite elements. Comput Fluids 35(4):372–392
6. Brackbill JU, Kothe DB, Zemach C (1992) A continuum method for modeling surface tension. J Comput Phys 100(2):335–354
7. Chen T, Minev PD, Nandakumar K (2004) A projection scheme for incompressible multiphase flow using adaptive Eulerian grid. Int J Numer Methods Fluids 45(1):1–19. doi:10.1002/fld.591
8. Clift R, Grace JR, Weber ME (2005) Bubbles, drops and particles. Dover, New York
9. Croce R, Griebel M, Schweitzer MA (2010) Numerical simulation of bubble and droplet deformation by a level set approach with surface tension in three dimensions. Int J Numer Methods Fluids 62(9):963–993. doi:10.1002/fld.2051
10. Damanik H (2011) Monolithic FEM techniques for viscoelastic fluids. PhD thesis, TU Dortmund, Dortmund
11. Di Pietro DA, Lo Forte S, Parolini N (2006) Mass preserving finite element implementations of the level set method. Appl Numer Math 56(9):1179–1195. doi:10.1016/j.apnum.2006.03.003
12. Donea J, Giuliani S, Laval H, Quartapelle L (1982) Finite element solution of the unsteady Navier-Stokes equations by a fractional step method. Comput Methods Appl Mech Eng 30(1):53–73
13. Dravid V, Songsermpong S, Xue ZJ, Corvalan CM, Sojka PE (2006) Two-dimensional modeling of the effects of insoluble surfactant on the breakup of a liquid filament. Chem Eng Sci 61(11):3577–3585

14. Dziuk G (1991) An algorithm for evolutionary surfaces. Numer Math 58(6):603–611
15. Farthing MW, Kees CE (2008) Implementation of discontinuous Galerkin methods for the level set equation on unstructured meshes. Technical note CHETN-XIII-2, U.S. Army Engineer Research and Development Center
16. Frolkovič P, Logashenko D, Wittum G (2008) Flux-based level set method for two-phase flow. Towards pure finite-volume discretization of incompressible two-phase flow using a level set formulation. In: Finite volumes for complex applications V. Wiley-ISTE, New York, pp 415–422
17. Ganesan S (2006) Finite element methods on moving meshes for free surface and interface flows. Published as a book by docupoint-Verlag, Magdeburg, 2006. PhD thesis, Otto-von-Guericke-Universität, Magdeburg
18. Ganesan S, Matthies G, Tobiska L (2007) On spurious velocities in incompressible flow problems with interfaces. Comput Methods Appl Mech Eng 196(7):1193–1202. doi:10.1016/j.cma.2006.08.018
19. Ganesan S, Tobiska L (2006) Computations on flows with interfaces using arbitrary Lagrangian Eulerian method. In: European conference on computational fluid dynamics ECCOMAS CFD 2006. TU Delft
20. Girault V, Raviart P-A (1986) Finite element methods for Navier-Stokes equations. Springer, Berlin
21. Glimm J, Grove JW, Li XL, Shyue K-M, Zeng Y, Zhang Q (1998) Three-dimensional front tracking. SIAM J Sci Comput 19(3):703–727
22. Gresho PM, Chan ST, Lee RL, Upson CD (1984) A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations. Part 1: Theory. Int J Numer Methods Fluids 4(6):557–598
23. Groß S, Reichelt V, Reusken A (2006) A finite element based level set method for two-phase incompressible flows. Comput Vis Sci 9(4):239–257
24. Groß S, Reusken A (2007) Finite element discretization error analysis of a surface tension force in two-phase incompressible flows. SIAM J Numer Anal 45(4):1679–1700
25. Hansbo A, Hansbo P (2002) An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. Comput Methods Appl Mech Eng 191(47–48):5537–5552
26. Houzeaux G, Codina R (2003) A Chimera method based on a Dirichlet/Neumann(Robin) coupling for the Navier-Stokes equations. Comput Methods Appl Mech Eng 192(31–32):3343–3377
27. Hu C, Shu C-W (1999) A discontinuous Galerkin finite element method for Hamilton-Jacobi equations. SIAM J Sci Comput 21(2):666–690
28. Hysing S (2006) A new implicit surface tension implementation for interfacial flows. Int J Numer Methods Fluids 51(6):659–672. doi:10.1002/fld.1147
29. Hysing S (2007) Numerical simulation of immiscible fluids with FEM level set techniques. PhD thesis, Universität Dortmund, Dortmund
30. Hysing S, Turek S (2005) The Eikonal equation: numerical efficiency vs. algorithmic complexity on quadrilateral grids. In: Proceedings of algorithmy, Bratislava, Slovak University of Bratislava, pp 22–31
31. Hysing S, Turek S, Kuzmin D, Parolini N, Burman E, Ganesan S, Tobiska L (2009) Quantitative benchmark computations of two-dimensional bubble dynamics. Int J Numer Methods Fluids 60(11):1259–1288. doi:10.1002/fld.1934
32. John V (2006) On the efficiency of linearization schemes and coupled multigrid methods in the simulation of a 3D flow around a cylinder. Int J Numer Methods Fluids 50(7):845–862. doi:10.1002/fld.1080
33. Jones MW, Bærentzen JA, Sramek M (2006) 3D distance fields: a survey of techniques and applications. IEEE Trans Vis Comput Graph 12(4):581–599
34. Kuzmin D, Turek S (2004) High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. J Comput Phys 198(1):131–158. doi:10.1016/j.jcp.2004.01.015
35. Li F, Shu C-W (2005) Reinterpretation and simplified implementation of a discontinuous Galerkin method for Hamilton-Jacobi equations. Appl Math Lett 18(11):1204–1209

36. Marchandise E, Remacle J-F (2006) A stabilized finite element method using a discontinuous level set approach for solving two phase incompressible flows. J Comput Phys 219(2):780–800
37. Matthies G (2002) Finite element methods for free boundary value problems with capillary surfaces. PhD thesis, Otto-von-Guericke-Universität, Magdeburg. Published as a book by Shaker Verlag, Aachen, 2002
38. Minev PD, Chen T, Nandakumar K (2003) A finite element technique for multifluid incompressible flow using Eulerian grids. J Comput Phys 187(1):255–273
39. Münster R, Mierka O, Turek S (2012) Finite element-fictitious boundary methods (FEM-FBM) for 3D particulate flow. Int J Numer Methods Fluids 69(2):294–313. doi:10.1002/fld.2558
40. Nagrath S, Jansen KE, Lahey RT (2005) Computation of incompressible bubble dynamics with a stabilized finite element level set method. Comput Methods Appl Mech Eng 194(42–44):4565–4587
41. Nguyen V-T, Peraire J, Khoo BC, Persson P-O (2010) A discontinuous Galerkin front tracking method for two-phase flows with surface tension. Comput Fluids 39(1):1–14. doi:10.1016/j.compfluid.2009.06.007
42. Nichols BD, Hirt CW (1975) Methods for calculating multidimensional, transient free surface flows past bodies. In: Proc. of the first international conf. on numerical ship hydrodynamics, Gaithersburg, MD
43. Osher S, Fedkiw R (2003) Level set methods and dynamic implicit surfaces. Springer, Berlin
44. Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. J Comput Phys 79(1):12–49. doi:10.1016/0021-9991(88)90002-2
45. Ouazzi A (2006) Finite element simulation of nonlinear fluids with application to granular material and powder. PhD thesis, Universität Dortmund, Dortmund, 2005. Published as a book by Shaker Verlag, Aachen
46. Owen HC (2009) A finite element model for free surface and two fluid flows on fixed meshes. PhD thesis, Universitat Politècnica de Catalunya, Barcelona
47. Parolini N (2004) Computational fluid dynamics for naval engineering problems. PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL)
48. Parolini N, Burman E (2005) A finite element level set method for viscous free-surface flows. In: Applied and industrial mathematics in Italy, proceedings of SIMAI 2004. World Scientific, Singapore, pp 416–427
49. Preußer T, Rumpf M (2002) A level set method for anisotropic geometric diffusion in 3D image processing. SIAM J Appl Math 62(5):1772–1793
50. Quecedo M, Pastor M (2001) Application of the level set method to the finite element solution of two-phase flows. Int J Numer Methods Eng 50(3):645–663
51. Ramaswamy B, Kawahara M (2005) Arbitrary Lagrangian-Eulerianc finite element method for unsteady, convective, incompressible viscous free surface fluid flow. Int J Numer Methods Fluids 7(10):1053–1075
52. Scardovelli R, Zaleski S (1999) Direct numerical simulation of free-surface and interfacial flow. Annu Rev Fluid Mech 31:567–603. doi:10.1146/annurev.fluid.31.1.567
53. Sethian JA (1999) Level set methods and fast marching methods: evolving interphases in computational geometry, fluid mechanics, computer vision, and materials science 2nd ed. Cambridge University Press, Cambridge
54. van Sint Annaland M, Deen NG, Kuipers JAM (2005) Numerical simulation of gas bubbles behaviour using a three-dimensional volume of fluid method. Chem Eng Sci 60(11):2999–3011. doi:10.1016/j.ces.2005.01.031
55. van Sint Annaland M, Dijkhuizen W, Deen NG, Kuipers JAM (2006) Numerical simulation of behavior of gas bubbles using a 3-D front-tracking method. AIChE J 52(1):99–110. doi:10.1002/aic.10607
56. Smolianski A (2001) Numerical modeling of two-fluid interfacial flows. PhD thesis, University of Jyväskylä

57. Smolianski A (2005) Finite-element/level-set/operator-splitting (FELSOS) approach for computing two-fluid unsteady flows with free moving interfaces. Int J Numer Methods Fluids 48(3):231–269
58. Sussman M, Ohta M (2009) A stable and efficient method for treating surface tension in incompressible two-phase flow. SIAM J Sci Comput 31(4):2447–2471
59. Tornberg A-K (2000) Interface tracking methods with applications to multiphase flows. PhD thesis, Royal Institute of Technology (KTH)
60. Turek S (1997) On discrete projection methods for the incompressible Navier-Stokes equations: an algorithmical approach. Comput Methods Appl Mech Eng 143(3–4):271–288. doi:10.1016/S0045-7825(96)01155-3
61. Turek S (1999) Efficient solvers for incompressible flow problems: An algorithmic and computational approach. Lecture notes in computational science and engineering, vol 6. Springer, Berlin
62. Turek S, Ouazzi A (2007) Unified edge-oriented stabilization of nonconforming FEM for incompressible flow problems: numerical investigations. J Numer Math 15(4):299–322
63. Turek S, Rivkind L, Hron J, Glowinski R (2006) Numerical study of a modified time-stepping θ-scheme for incompressible flow simulations. J Sci Comput 28(2–3):533–547. doi:10.1007/s10915-006-9083-y
64. Turek S, Schäfer M (1996) Benchmark computations of laminar flow around cylinder. In: Flow simulation with high-performance computers II. Notes on numerical fluid mechanics, vol 52. Vieweg, Wiesbaden, pp 547–566
65. Unverdi SO, Tryggvason G (1992) A front-tracking method for viscous, incompressible, multi-fluid flows. J Comput Phys 100(1):25–37. doi:10.1016/0021-9991(92)90307-K
66. Veneziani A, Villa U (2011) ALADINS: an ALgebraic splitting time ADaptive solver for the incompressible Navier-Stokes equations. Part 1: basic settings and analysis. Technical report TR-2011-010, Emory University
67. Welch JE, Harlow FH, Shannon JP, Daly BJ (1966) The MAC method: a computing technique for solving viscous, incompressible, transient fluid-flow problems involving free surfaces. Los Alamos Scientific Laboratory Report LA-3425
68. Yu J-D, Sakai S, Sethian JA (2007) Two-phase viscoelastic jetting. J Comput Phys 220(2):568–585. doi:10.1016/j.jcp.2006.05.020
69. Zhao H (2005) A fast sweeping method for Eikonal equations. Math Comput 74(250):603–627

Chapter 5
GAs and Nash GAs Using a Fast Meshless Method for CFD Design

Hong Wang, Hong-Quan Chen, and Jacques Periaux

Abstract Solving CFD inverse problems dealing with complex aerodynamic configurations like multi-element airfoils remains a difficult and expensive procedure, which requires seamless interfacing between several softwares like computer-aided design (CAD) system, mesh generator, flow analyzer, and an optimizer. It is essential to ensure the mesh quality during the optimization procedure for maintaining an accurate design. A fast meshless method using second and fourth order artificial dissipations and dynamic clouds of points based on the Delaunay graph mapping strategy is introduced to solve inverse computational fluid dynamics problems. The purpose of this paper is to use genetic algorithms and Nash genetic algorithms for position reconstructions of oscillating airfoils. The main feature of this paper is a detailed investigation on inverse problems in aerodynamics using both flexibility and efficiency of the fast meshless method. Comparisons of prescribed and computed aerodynamics parameters are presented for position reconstruction problems in aerodynamic design using both the fast meshless method coupled with artificial dissipation and a finite volume method. Numerical results are presented to illustrate the potential of the fast meshless method coupled with artificial dissipation and evo-

H. Wang (✉) · J. Periaux
Department of Mathematical Information Technology, University of Jyväskylä,
P.O. Box 35 (Agora), 40014 Jyväskylä, Finland
e-mail: hong.m.wang@jyu.fi

H.-Q. Chen
Department of Aerodynamics, Nanjing University of Aeronautics and Astronautics, 29 Yudao Jie,
Nanjing 210016, P.R. China
e-mail: hqchenam@nuaa.edu.cn

J. Periaux
International Center for Numerical Methods in Engineering (CIMNE), Edificio C1, Gran Captian,
s/n, 08034 Barcelona, Spain
e-mail: jperiaux@gmail.com

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_5,

lutionary algorithms, to solve more complex optimization problems of industrial interest occurring in multidisciplinary design.

5.1 Introduction

Compared to direct computational fluid dynamics (CFD) problems, inverse problems [15] have been of ongoing interest to aerodynamic researchers. The position reconstruction is one of the important problems in high lift devices using multi element airfoils configurations. The goal of our present study is to implement efficiently on the computer a simple reconstruction problem with Genetic Algorithms (GAs) [10] and/or Nash GAs [11, 15] to reconstruct the target position of oscillating airfoils based on prescribed conditions.

Meshless methods (see, e.g., [2–9]) do not use the concept of mesh topology and provide more geometrical flexibility for computing flow fields. In addition, they are also useful in design optimization problems around complex configurations without constraints required by mesh quality and topology. A fast meshless method coupled with artificial dissipation (AD) using second and fourth order derivatives is employed for solving two-dimensional (2D) Euler equations. Spatial derivatives of the governing equations are approximated by a weighted least square (WLS) method discretizing the computational domain into clouds of points (see, e.g., [1–4]). An explicit five-stage Runge-Kutta scheme is utilized to reach the steady-state solution. A local time-stepping method and a residual averaging [3] are employed to accelerate the rate of convergence. Dynamic clouds of points based on the Delaunay graph mapping [8] are selected to ensure the flow field points following the movements of body boundaries.

The proposed approach is validated by comparing our numerical results against a finite volume method presented in [6] for a single oscillating NACA0012 airfoil. In this paper, we have tested the position reconstructions of oscillating airfoils operating in transonic regimes for aerodynamic design. Position reconstruction of a single airfoil has been tested using GAs optimizer. Position reconstruction of two airfoils has been tested with Nash GAs using both the fast meshless method coupled with AD and the finite volume method on the same computational nodes. Comparisons of prescribed and computed parameters are presented to show the efficacy of the fast meshless method coupled with AD and Nash game strategy in the position reconstruction problems in aerodynamic design.

The rest of the paper is organized as follows. Section 2 describes the methodology of the dynamic cloud method based on the Delaunay graph mapping strategy and the meshless method coupled with AD. Section 3 shows the validation of the proposed meshless method. Section 4 conducts two practical optimization applications and conclusions are presented in Sect. 5.

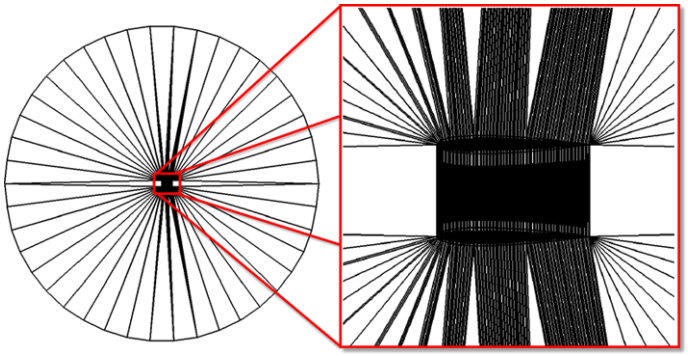


Fig. 5.1 Global and close-up views of a Delaunay graph in the case of NACA0012 airfoils

5.2 Methodology

5.2.1 Dynamic Cloud Method Based on the Delaunay Graph Mapping Strategy

In order to simulate the relative movement of boundaries in the position reconstruction, it is required that a cloud of points has the ability to move with the rigid body boundaries. Hence, a fast and efficient dynamic cloud method based on the Delaunay graph mapping strategy [8] is introduced here.

Firstly, as shown in Fig. 5.1, a Delaunay triangulation of the computational field is set up by using the given points located on the boundaries for BI-NACA0012 airfoils. Then, the triangulation is contained for every point $P(x, y)$ in the computational field. Notate the points of every element $E(x_1, y_1)$, $E(x_2, y_2)$ and $E(x_3, y_3)$, then the coordinates of the point can be expressed as

$$\begin{cases} x = a_1 x_1 + a_2 x_2 + a_3 x_3, \\ y = a_1 y_1 + a_2 y_2 + a_3 y_3, \end{cases} \tag{5.1}$$

where $a_1 = S_1/S$, $a_2 = S_2/S$, $a_3 = S_3/S$; S, S_1, S_2, S_3 are the relevant triangle's areas [8]. Then, all the background points by the movement of the boundary's points are adjusted. The coordinates of the relevant triangle become $E(x_1', y_1')$, $E(x_2', y_2')$ and $E(x_3', y_3')$, and the new coordinates of point can be denoted as

$$\begin{cases} x' = a_1 x_1' + a_2 x_2' + a_3 x_3', \\ y' = a_1 y_1' + a_2 y_2' + a_3 y_3'. \end{cases} \tag{5.2}$$

In [14] it is shown that better results can be obtained by using the Delaunay graph mapping strategy to ensure the flow field points following the movements of the body boundaries without any iteration. And compared to the spring analogy method described in [5].

5.2.2 Governing Equations

The so-called Euler equations represent the conservation principle for mass, momentum, and energy for inviscid fluids. In a 2D Cartesian coordinate system, Euler equations are expressed in the following form:

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} = 0, \tag{5.3}$$

where t denotes time and (x, y) the Cartesian coordinates. The expressions of conservative variables $\mathbf{W}$ and convective fluxes $\mathbf{E}$, $\mathbf{F}$ are introduced as

$$\mathbf{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, \qquad \mathbf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e+p)u \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (e+p)v \end{bmatrix}, \tag{5.4}$$

where ρ denotes the density, u is the x-velocity component, v is the y-velocity component, p is the pressure, and e is the total energy per unit volume. For an ideal gas, e can be written as

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho\left(u^2 + v^2\right),$$

where γ is the ratio of specific heat. Additionally, the equation of state is given by

$$p = \rho \bar{R} T,$$

where T is the static temperature and $\bar{R}$ is the ideal gas constant.

5.2.3 Spatial Discretization

The WLS method [4] is used to approximate the spatial first-order derivatives, and in the cloud $C(i)$ as shown in Fig. 5.2, (5.3) becomes

$$\left.\frac{\partial \mathbf{W}}{\partial t}\right|_i + \left(\frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y}\right)_i = 0. \tag{5.5}$$

For the convective fluxes, let

$$\mathbf{Q}_i = \left(\frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y}\right)_i. \tag{5.6}$$

According to the WLS method [4], (5.6) could be written as

$$\mathbf{Q}_i = \sum \alpha_{ik} \mathbf{E}_{ik} + \sum \beta_{ik} \mathbf{F}_{ik}. \tag{5.7}$$

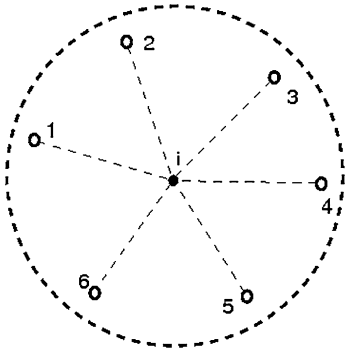


Fig. 5.2 A typical structure of the cloud $C(i)$

Then the governing equation can be written as

$$\frac{d\mathbf{W}_i}{dt} = -\left(\mathbf{Q}_i - \sum_{k=1}^{N} d_{ik}\right), \tag{5.8}$$

where [3]

$$\begin{aligned}
d_{ik} &= \epsilon_{ik}^{(2)}(\mathbf{W}_k - \mathbf{W}_i) - \epsilon_{ik}^{(4)}\left(\nabla^2\mathbf{W}_k - \nabla^2\mathbf{W}_i\right),\\
\epsilon_{ik}^{(2)} &= K^{(2)}\lambda_{ik}\max(\upsilon_i, \upsilon_k),\\
\epsilon_{ik}^{(4)} &= \lambda_{ik}\max\left[0, K^{(4)} - \epsilon_{ik}^{(2)}\right],\\
\upsilon_i &= \frac{|\nabla^2 P_i|}{\sum_{k=1}^{N}(P_i + P_k)},\\
\nabla^2\mathbf{W}_i &= \sum_{k=1}^{N}\mathbf{W}_k - N\mathbf{W}_i,\\
\lambda_{ik} &= |\alpha_{ik}u + \beta_{ik}v| + c\sqrt{\alpha_{ik}^2 + \beta_{ik}^2}.
\end{aligned}$$

Here $c = \sqrt{\gamma p/\rho}$ is the local speed of sound.

5.2.4 Temporal Discretization

Within the cloud $C(i)$, the semi-discretisation Euler equations are rewritten as

$$\left.\frac{\partial \mathbf{W}}{\partial t}\right|_i = \mathbf{R}_i, \tag{5.9}$$

where $\mathbf{R}_i$ means the residual value. An explicit scheme is used for time discretisation in (5.9), and we get

$$\frac{\mathbf{W}_i^{n+1} - \mathbf{W}_i^n}{\Delta t} = \mathbf{R}_i. \tag{5.10}$$

The superscripts n and $(n+1)$ denote the time levels. Hence, $\mathbf{W}^n$ means the flow solution at the present time t, and $\mathbf{W}^{n+1}$ represents the solution at the time $(t + \Delta t)$. An explicit five-stage Runge-Kutta time integration scheme is used

$$\begin{cases} \mathbf{W}_i^{(0)} = \mathbf{W}_i^n, \\ \mathbf{W}_i^{(1)} = \mathbf{W}_i^{(0)} + \alpha_1 \Delta t_i \mathbf{R}_i^{(0)}, \\ \mathbf{W}_i^{(2)} = \mathbf{W}_i^{(0)} + \alpha_2 \Delta t_i \mathbf{R}_i^{(1)}, \\ \mathbf{W}_i^{(3)} = \mathbf{W}_i^{(0)} + \alpha_3 \Delta t_i \mathbf{R}_i^{(2)}, \\ \mathbf{W}_i^{(4)} = \mathbf{W}_i^{(0)} + \alpha_4 \Delta t_i \mathbf{R}_i^{(3)}, \\ \mathbf{W}_i^{(5)} = \mathbf{W}_i^{(0)} + \alpha_5 \Delta t_i \mathbf{R}_i^{(4)}, \\ \mathbf{W}_i^{n+1} = \mathbf{W}_i^{(5)}, \end{cases} \tag{5.11}$$

where α_k, $k = 1, 2, 3, 4, 5$, represents the stage coefficients, and we have $\alpha_1 = \frac{1}{4}$, $\alpha_2 = \frac{1}{6}$, $\alpha_3 = \frac{3}{8}$, $\alpha_4 = \frac{1}{2}$, $\alpha_5 = 1$.

The major disadvantage of the explicit scheme is that the time step is restricted by the Courant-Friedrichs-Lewy (CFL) stability condition [3].

5.2.5 Acceleration Techniques

In order to accelerate the convergence, a local time stepping method and an implicit residual averaging method are employed in our present work. The local time step Δt_i of a discrete point is given by the equation [3, 12, 13]

$$\Delta t = \frac{C_{\mathrm{CFL}}}{\sum_{k=1}^{N} |\alpha_{ik} u + \beta_{ik} v| + c\sqrt{\alpha_{ik}^2 + \beta_{ik}^2}}, \tag{5.12}$$

where C_{CFL} denotes the coefficient of CFL.

In the meshless method for the time marching equation, let $\mathbf{R}_i$ represent the residual at node i. The new residual [3] can be given as

$$\mathbf{R}_i' = \frac{\mathbf{R}_i + \epsilon \sum_{k=1}^{M} \mathbf{R}_k'}{1 + \epsilon M}, \tag{5.13}$$

where $\epsilon = [0.2, 0.5]$ and it can be obtained by performing two Jacobi iterations. The above technique allows the CFL number to be increased twofold or threefold when

compared to the unsmoothed value, and consequently the CFL number is increased from $2\sqrt{2}$ to 5 in the present study.

5.3 Validation of the Fast Meshless Method Implemented with Artificial Dissipation (AD)

In order to validate the proposed fast meshless method coupled with AD, a single NACA0012 airfoil operating with flow conditions at a 3.0° angle of attack and a Mach number 0.5 is tested. Figure 5.3 provides both the global view and the close-up view of the cloud of points distributed around one single NACA0012 airfoil, and Fig. 5.4 shows both the global view and the close-up view of the mesh distributed for the same airfoil. There are 5047 nodes in the whole computational domain in the meshless method and 9762 elements in the mesh method. Figure 5.5 shows the comparison of surface pressure coefficients for this test case using the fast meshless method coupled with AD and the finite volume method in [6].

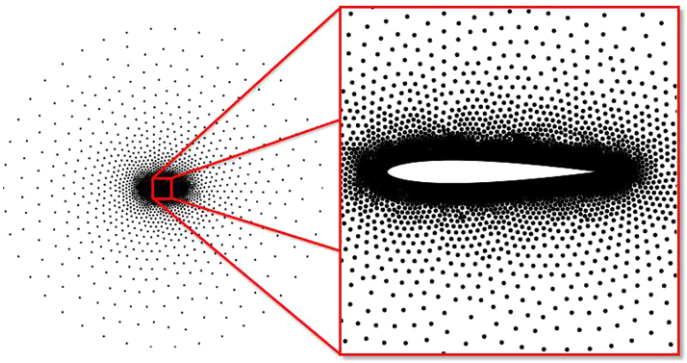


Fig. 5.3 Global and close-up views of the cloud of points for the NACA0012 airfoil

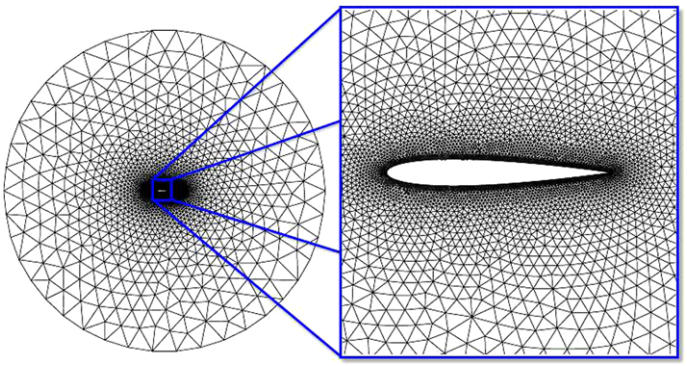


Fig. 5.4 Global and close-up views of the mesh for the NACA0012 airfoil

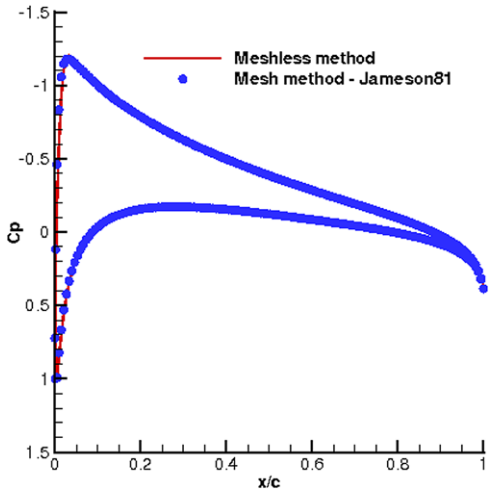


Fig. 5.5 Comparisons of surface pressure coefficients for the NACA0012 airfoil

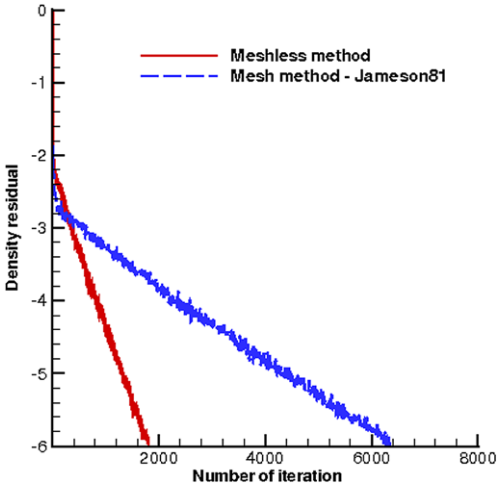


Fig. 5.6 Comparisons of the convergence history for the NACA0012 airfoil

Figure 5.6 shows the comparison of the convergence history for this case using the fast meshless method coupled with AD and the referenced mesh method [6]. As shown in the histogram in Figs. 5.7 and 5.8, the meshless method coupled with AD in this test case saves 71.5 % in the iteration costs compared to the finite volume method described in [6]. In terms of the CPU time needed in this test case, the meshless method coupled with AD saves 65.7 % compared to the finite volume method in [6]. The computer hardware used in this paper is an Intel(R) Core(TM) with 2 Quad CPU Q9650 with frequency 2.00 GHz/2.99 GHz and 3.00 GB of RAM.

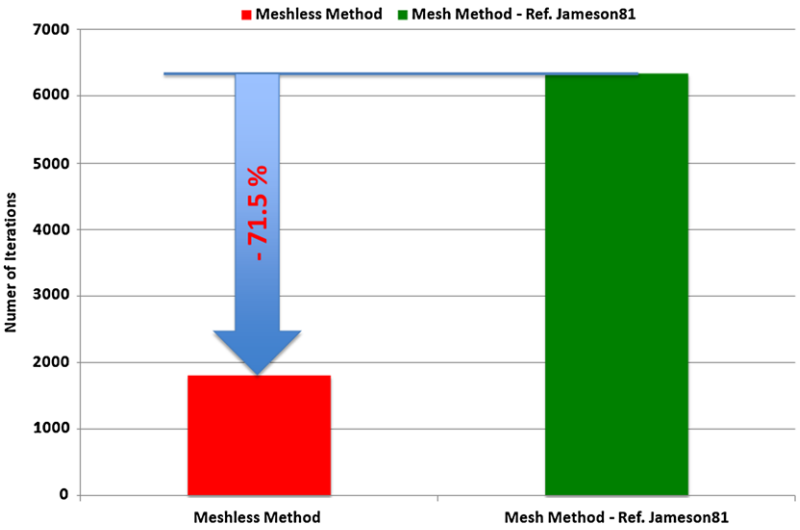


Fig. 5.7 Comparisons of the convergence history in terms of the number of iterations for the NACA0012 airfoil

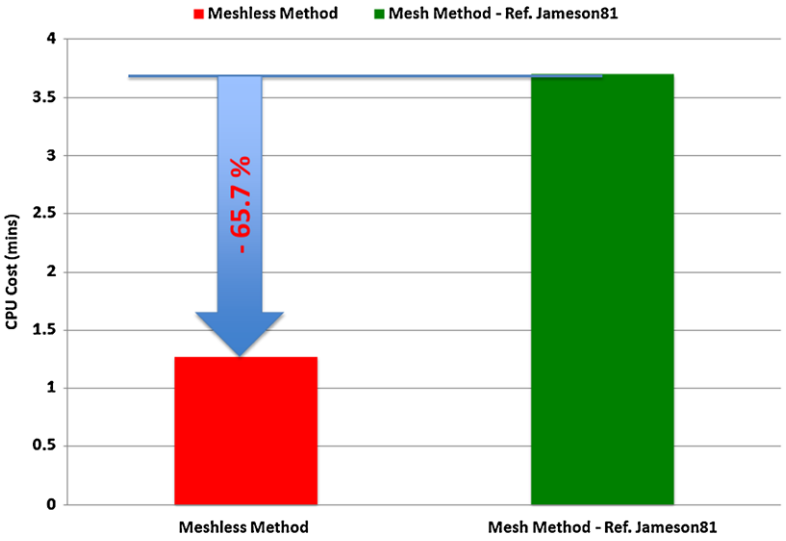


Fig. 5.8 Comparisons of the convergence history in terms of the CPU cost for the NACA0012 airfoil

5.4 Practical Optimization Applications

In this section, both the fast meshless method coupled with AD and the finite volume method referenced in [6] are used to test two inverse position reconstruction problems: a single pitching NACA0012 airfoil and BI-NACA0012 airfoils.

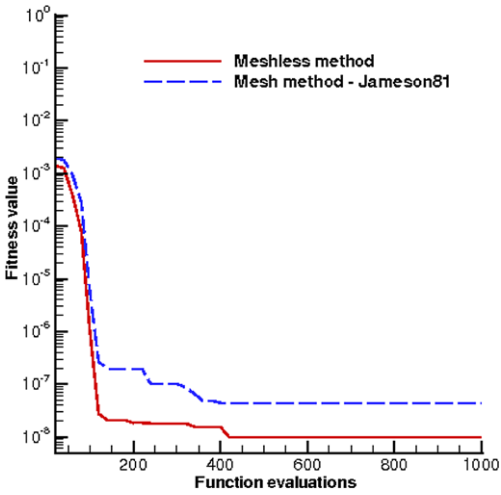


Fig. 5.9 Comparisons of the convergence history in terms of fitness value for a single NACA0012 airfoil

5.4.1 A Single Pitching NACA0012 Airfoil

Let one airfoil oscillate in pitch about its quarter chord, and the rotating angle α is selected as the design parameter. The objective function is defined according to the surface pressure coefficients as

$$\min f(\alpha) = \sum_{i=1}^{M} \left| C_p(\alpha) - C_p\left(\alpha^*\right) \right|_i^2, \tag{5.14}$$

where M is the total number of points distributed on the surface of an airfoil, the search space is $\alpha \in [-10.0°, 10.0°]$, and α^* is the prescribed design variable. Parameters of the GAs optimizer are chosen as: the size of population is 20, the probability of crossover is 0.85, and the probability of mutation is 0.01.

The flow conditions of the reconstruction test case are as follows: the Mach number is 0.8 and the target angle of attack is 0.0°. Figure 5.9 shows the convergence history of fitness value during the reconstruction process using the fast meshless method coupled with AD and the finite volume method in [6] separately. As shown on the histogram of Fig. 5.10, the meshless method coupled with AD saves 69.7 % compared to the finite volume method referenced in [6] in terms of the CPU time cost.

5.4.2 BI-NACA0012 Airfoil's Configuration

Let two airfoils oscillate in pitch about their quarter chords, and rotating angles α_1, α_2 are selected as design parameters. The two objective functions defined in a

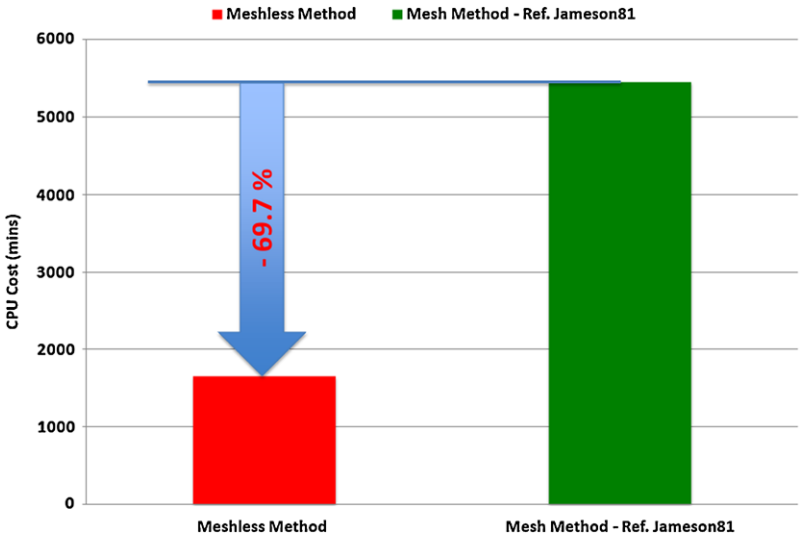


Fig. 5.10 Comparisons of the convergence history in terms of the CPU cost for the NACA0012 airfoil

reconstruction problem solved by the Nash-GAs optimizer are

$$\min f_1(\alpha_1, \alpha_2^{**}) = \sum_{i=1}^{M_1} \left| C_p(\alpha_1) - C_p(\alpha_1^*) \right|_i^2 + \sum_{i=1}^{M_2} \left| C_p(\alpha_2^{**}) - C_p(\alpha_2^*) \right|_i^2, \quad (5.15)$$

$$\min f_2(\alpha_1^{**}, \alpha_2) = \sum_{i=1}^{M_1} \left| C_p(\alpha_1^{**}) - C_p(\alpha_1^*) \right|_i^2 + \sum_{i=1}^{M_2} \left| C_p(\alpha_2) - C_p(\alpha_2^*) \right|_i^2, \quad (5.16)$$

where M_1 is the total number of points distributed on the surface of the upper airfoil while M_2 is the total number of points distributed on the surface of the lower airfoil, the search spaces are $\alpha_1 \in [-10.0°, 10.0°]$, $\alpha_2 \in [-10.0°, 10.0°]$, and α_1^*, α_2^* are prescribed parameters. The parameters in Nash GAs are chosen as follows: the size of the population is 10, the probability of crossover is 0.85, and the probability of mutation is 0.02.

The Euler flow conditions around the BINACA0012 configuration are the following: the Mach number is 0.5 and the prescribed parameters are 0.0° and 0.0°. Figure 5.11 shows the convergence history of the objective function during the reconstruction process using Nash GAs based on the meshless method coupled with AD and the finite volume method in [6]. As shown in the histogram in Fig. 5.12, the meshless method coupled with AD saves 75.8 % compared to the finite volume method in [6] in terms of the CPU time cost.

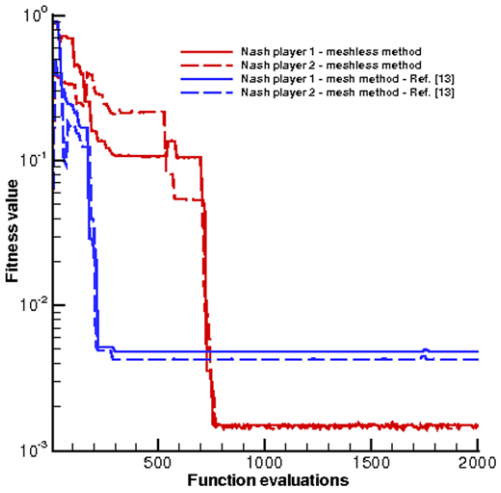


Fig. 5.11 Comparisons of the convergence history for BINACA0012 airfoils in terms of the objective function using the fast meshless method coupled with AD and the standard mesh method

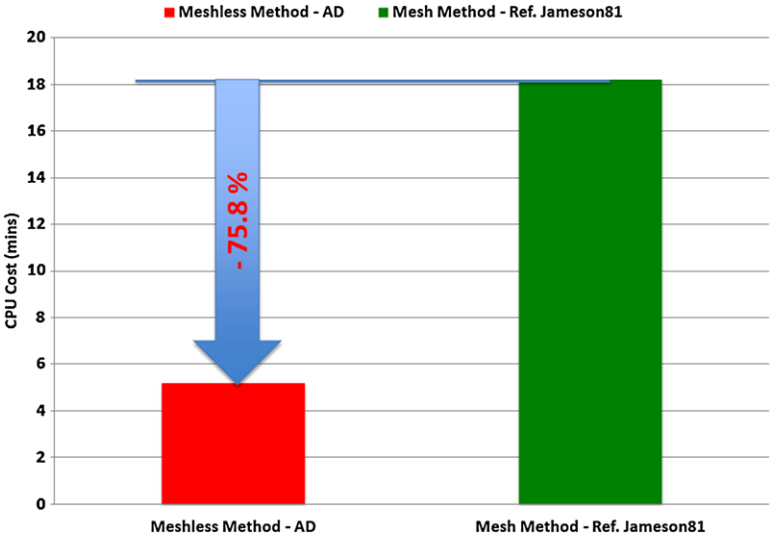


Fig. 5.12 Comparisons of the convergence history in terms of the CPU cost for the BI-NACA0012 airfoils

5.5 Conclusions and Future

A fast Euler meshless method using artificial dissipations is used in this paper. Dynamic clouds of points based on a Delaunay graph mapping strategy have been introduced to ensure that flow field points can easily follow the movements of solid body boundaries. Position reconstructions of oscillating airfoils operating in transonic regimes have been tested for future aerodynamic design like high lift devices. First a single airfoil position reconstruction has been tested successfully with a sim-

ple GAs optimizer. Then, position reconstruction of two airfoils has been tested with Nash GAs using both the fast meshless method coupled with AD and the finite volume method referenced in [6] using the same number of computational nodes. Comparisons of target geometries and computed parameters are presented to prove the superiority of the meshless Euler flow analyzer methods implemented with AD coupled with the Nash evolutionary optimizer for position reconstruction inverse problems in aerodynamic design. This study is a roadmap to more complex design optimization problems which can benefit of game coalitions [7] in terms of accuracy and efficiency.

References

1. Batina JT (1992) A fast implicit upwind solution algorithm for three-dimensional unstructured dynamics meshes. AIAA paper 1992-447
2. Batina JT (1993) A gridless Euler/Navier-Stokes solution algorithm for complex-aircraft applications. AIAA paper 1993-333
3. Blazek J (2001) Computational fluid dynamics: principles and applications. Elsevier, Amsterdam
4. Chen HQ (2003) Implicit gridless method for Euler equations. Chin J Comput Phys 20(1):9–13
5. Farhat C, Degand C, Koobus B, Lesoinne M (1998) An improved method of spring analogy for dynamic unstructured fluid meshes. AIAA paper 1998-2070
6. Jameson A, Schmidt W, Turkel E (1981) Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. In: AIAA 14th fluid and plasma dynamics conference, Palo Alto, CA, 1981
7. Lee DS, Gonzalez LF, Periaux J, Srinivas K (2011) Efficient hybrid-game strategies coupled to evolutionary algorithms for robust multidisciplinary design optimization in aerospace engineering. IEEE Trans Evol Comput 15(2):133–150
8. Liu XQ, Qin N, Xia H (2006) Fast dynamic grid deformation based on Delaunay graph mapping. Chin J Comput Phys 211(2):405–423
9. Ma ZH, Chen HQ, Wu XJ (2006) A gridless-finite volume hybrid algorithm for Euler equations. Chin J Aeronaut 19(4):286–294
10. Michalewicz Z (1992) Genetic algorithms + data structures = evolution programs. Springer, Berlin
11. Nash J (1951) Non-cooperative games. Ann Math 54:286–295
12. Pulliam TH (1986) Artificial dissipation models for the Euler equations. AIAA J 24(12):1931–1940
13. Pulliam TH, Steger JL (1985) Recent improvements in efficiency, accuracy, and convergence for implicit approximate factorization algorithms. AIAA paper 1985-360
14. Wang H, Chen HQ, Periaux J (2010) A study of gridless method with dynamic clouds of points for solving unsteady CFD problems in aerodynamics. Int J Numer Methods Fluids 64(1):98–118
15. Wang JF, Wu YZ, Periaux J (2002) Genetic algorithms and game theory for high lift design problems in aerodynamics. Trans Nanjing Univ Aeronaut Astronaut 19(1):7–13

Part II
Reliable Methods for Computer Simulation

Chapter 6
Balancing Discretization and Iteration Error in Finite Element A Posteriori Error Analysis

Rolf Rannacher and Jevgeni Vihharev

Abstract This article surveys recent developments in a combined a posteriori analysis for the discretization and iteration errors in the finite element approximation of elliptic PDE systems. The underlying theoretical framework is that of the **D**ual **W**eighted **R**esidual (DWR) method for goal-oriented error control. Based on computable a posteriori error estimates the algebraic iteration can be adjusted to the discretization in a successive mesh adaptation process. The performance of the proposed method is demonstrated for several model situations including the simple Poisson equation, the Stokes equations in fluid mechanics and the KKT system of a linear-quadratic elliptic optimal control problem. Furthermore, extensions are discussed for certain classes of nonlinear problems including eigenvalue problems and nonlinear reaction-diffusion equations.

6.1 Introduction

The use of adaptive techniques based on a posteriori estimates for the discretization error is well accepted in the context of finite element discretization of partial differential equations (see, e.g., [1, 8, 25]). Although the convergence properties of linear as well as nonlinear iterative methods such as the multigrid method or the Newton method are discussed in many publications (see, e.g., [3, 10–12, 14]), there are only few results on a posteriori error estimation of the iteration error. In the case of solving the Poisson equation, work has been done in [6] and was extended to the Stokes equations in [4]. There, the automatic control of the discretization and multigrid errors has been developed with respect to L^2- and energy norms. The reliability of the proposed adaptive algorithm has been verified on uniformly refined meshes.

However, in many applications, the error measured in global norms does not provide useful bounds for the error in terms of a given functional, a so-called *quantity*

R. Rannacher (✉) · J. Vihharev
Institute of Applied Mathematics, University of Heidelberg, INF 293/294, 69120 Heidelberg, Germany
e-mail: rolf.rannacher@iwr.uni-heidelberg.de

J. Vihharev
e-mail: jevgeni.vihharev@iwr.uni-heidelberg.de

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_6,

of interest. In this work, we propose the simultaneous control of both discretization and iteration errors with respect to a prescribed output functional. This approach is based on a posteriori error estimation by dual weighted residuals as presented in [8] as the **D**ual **W**eighted **R**esidual (DWR) method. We incorporate the adaptive iteration method into the solution process of a given problem. It seems natural to stop the linear or nonlinear iteration when the error due to the approximate solution of the discrete equations is comparable to the error due to the finite element discretization itself. To this purpose, we derive an a posteriori error estimator which simultaneously assesses the influences of the discretization and the inexact solution of the arising algebraic equations. This allows us to balance both sources of errors.

For illustration, we consider the model problem

$$Au = f \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \Gamma, \tag{6.1}$$

with a linear elliptic operator A and a right-hand side $f \in L^2(\Omega)$ where Ω is assumed to be a bounded domain in $\mathbb{R}^d$, $d \in \{2, 3\}$, with polygonal respectively polyhedral boundary Γ. For simplicity, we impose homogeneous Dirichlet boundary conditions. However, the techniques developed in this paper can also be applied to problems with other types of boundary conditions. For the variational formulation of the problem (6.1), we introduce the Hilbert space $V := H_0^1(\Omega)$ and the L^2-scalar product $(v, w) := (v, w)_{L^2(\Omega)}$. With the bilinear form $a(\cdot, \cdot) \colon V \times V \to \mathbb{R}$ associated to the linear operator A, the weak formulation of the problem (6.1) reads as follows: Find $u \in V$ such that

$$a(u, \phi) = (f, \phi) \quad \forall \phi \in V. \tag{6.2}$$

We discretize this problem by a standard finite element method (see [13]) in finite dimensional spaces $V_h \subset V$ resulting in "discrete" problems

$$a(u_h, \phi_h) = (f, \phi_h) \quad \forall \phi_h \in V_h, \tag{6.3}$$

which are equivalent to linear systems of algebraic equations. Usually the a posteriori error estimators for the discretization error $u - u_h$ are derived under the assumption that the discrete problems (6.3) are solved exactly. This ensures the crucial property of the Galerkin orthogonality,

$$a(u - u_h, \phi_h) = 0, \quad \phi_h \in V_h. \tag{6.4}$$

In contrast, here, we assume that the discrete problems are solved only approximately and denote the obtained approximate solution in V_h by $\tilde{u}_h$ in contrast to the notation u_h for the "exact" discrete solution. Let the quantity of interest $J(u)$ of the computation be given in terms of a linear functional $J \colon V \to \mathbb{R}$. Our goal is the derivation of an a posteriori error estimate of the form

$$\left| J(u) - J(\tilde{u}_h) \right| \le \eta_h + \eta_{\text{it}}. \tag{6.5}$$

Here, η_h and η_{it} denote error estimators which can be evaluated from the computed discrete solution $\tilde{u}_h$, where η_h assesses the error due to the finite element

discretization and η_{it} the error due to the inexact solution of the discrete equations. The adaptation strategy then aims at equilibrating these two error components, $\eta_{\text{it}} \approx \eta_h \approx \frac{1}{2}\text{TOL}$, according to the prescribed error tolerance TOL. This results in a practical stopping criterion for the linear or nonlinear algebraic iteration.

This article is based on the results of the articles [6, 17, 20, 21]. The outline is as follows: In Sect. 6.2, we describe the finite element discretization of the problem (6.1) and develop the principles of the DWR method for goal-oriented a posteriori error estimation of the discretization as well as the iteration errors. Section 6.2.1 discusses the practical evaluation of these error estimators and the implementation of the resulting adaptation strategies. The numerical results presented in Sect. 6.2.2 demonstrate the efficiency and reliability of the proposed method for a prototypical scalar model problem. In Sect. 6.3 this approach is developed for the associated symmetric eigenvalue problem. Then, Sect. 6.4 is devoted to the treatment of different types of saddle point problems, the Stokes system in fluid mechanics, and the Karush-Kuhn-Tucker (KKT) system in linear-quadratic optimization. Finally, in Sect. 6.5, we consider the extension of our theory to the Newton iteration for nonlinear elliptic problems. The article concludes with Sect. 6.6, which addresses current work and open problems.

6.2 Goal-Oriented Mesh Adaptation: The DWR Method

We briefly sketch the essentials of "goal-oriented" a posteriori error estimation and mesh adaptation underlying the **D**ual **W**eighted **R**esidual (DWR) method [2, 7, 8].

Let the goal of the computation be the approximation of a scalar quantity $J(u)$ with maximal accuracy TOL on a mesh $\mathbb{T}_h$ from models

$$\mathscr{A}(u) = 0, \qquad \mathscr{A}_h(u_h) = 0.$$

In this process the goal of adaptivity is the optimization of the mesh $\mathbb{T}_h$ guided by an a posteriori error estimate of the form

$$J(u) - J(u_h) \approx \eta(u_h) := \sum_{K \in \mathbb{T}_h} \rho_K(u_h)\omega_K$$

with local cell residuals $\rho_K(u_h)$ and weights ω_K (sensitivity factors). Then, the mesh adaptation is driven by the local error indicators $\eta_K := \rho_K(u_h)\omega_K$. The inherent problem in this approach is that usually only an approximation $\tilde{u}_h$ of the exact discrete solution u_h is available obtained by a nonlinear or linear iteration process.

For illustration, we consider the following model situation. For the solution of the boundary value problem

$$-\Delta u = f \quad \text{in } \Omega \subset \mathbb{R}^2,\ u_{|\partial\Omega} = 0, \tag{6.6}$$

the quantity $J(u)$ is to be determined, where $J(\cdot)$ is a linear functional defined on the natural solution space of this problem. The variational formulation of (6.6) reads

$$u \in V: \quad a(u, \psi) := (\nabla u, \nabla \psi) = (f, \psi) \quad \forall \psi \in V, \tag{6.7}$$

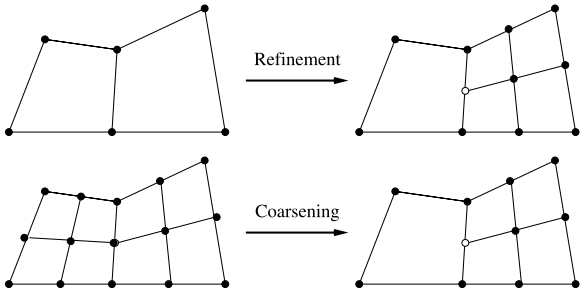


Fig. 6.1 Mesh refinement and coarsening using "hanging nodes"

where $V := H_0^1(\Omega)$ is the usual first-order Sobolev Hilbert space. For approximating this variational problem, we consider a Galerkin finite element method using subspaces $V_h \subset V$ (P_1 or Q_1 elements):

$$u_h \in V_h: \quad a(u_h, \psi_h) = (f, \psi_h) \quad \forall \psi_h \in V_h. \tag{6.8}$$

The spaces V_h are defined on form-regular decompositions $\mathbb{T}_h = \{K\}$ of $\overline{\Omega}$ consisting of closed cells K (triangular/quadrilateral in 2D and tetrahedral/hexahedral in 3D) with diameter h_K (see [13]). The global mesh size is $h := \max_{K\in\mathbb{T}_h} h_K$. To ease local mesh adaptation, we allow "hanging nodes" (at most one per face or edge) where the corresponding "irregular" nodal values are eliminated from the system by linear interpolation of neighboring regular nodal values (see Fig. 6.1).

The error $e := u - u_h$ satisfies the Galerkin orthogonality relation

$$a(e, \psi_h) = 0, \quad \psi_h \in V_h. \tag{6.9}$$

We introduce the associated continuous and discrete "dual" problems

$$z \in V: \quad a(\phi, z) = J(\phi) \quad \forall \phi \in V, \tag{6.10}$$

$$z_h \in V_h: \quad a(\phi_h, z_h) = J(\phi_h) \quad \forall \phi_h \in V_h. \tag{6.11}$$

Taking the test function $\phi = e$ in (6.10) yields the error identity

$$J(e) = a(e, z) = a(e, z - \psi_h) = (f, z - \psi_h) - a(u_h, z - \psi_h) =: \rho(u_h)(z - \psi_h)$$

with an arbitrary $\psi_h \in V_h$. By cell-wise integration by parts, we obtain

$$J(e) = \sum_{K\in\mathbb{T}_h} \big\{ \big(R(u_h), z - \psi_h\big)_K + \big(r(u_h), z - \psi_h\big)_{\partial K} \big\},$$

with the cell and edge residuals $R(u_h)$ and $r(u_h)$ defined by

$$R(u_h)_{|K} := f + \Delta u_h, \qquad r(u_h)_{|\Gamma} := \begin{cases} -\frac{1}{2} n \cdot [\nabla u_h], & \text{if } \Gamma \subset \partial K \setminus \partial\Omega, \\ 0, & \text{if } \Gamma \subset \partial\Omega, \end{cases}$$

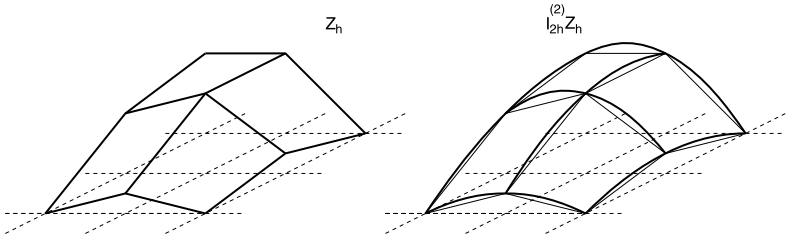


Fig. 6.2 Local post-processing by higher-order patchwise interpolation: "biquadratic" interpolation of computed "bilinear" nodal values

where $[\nabla u_h]$ denotes the jump of the normal derivative across interelement edges. Then, using the refinement indicators

$$\eta_K := \left| \big(R(u_h), z - \psi_h\big)_K + \big(r(u_h), z - \psi_h\big)_{\partial K} \right|,$$

the mesh adaptation aims at "error balancing", i.e.,

$$\eta := \sum_{K \in \mathbb{T}_h} \eta_K, \qquad N := \#\{K \in \mathbb{T}_h\}, \quad \eta_K \approx \mathrm{TOL}/N,$$

which at the end results in $\eta \approx \mathrm{TOL}$.

The unknown dual solution z occurring in the error indicators η_K is approximated by local higher-order post-processing from the computed dual solution z_h,

$$z - I_h^{(1)} z \;\approx\; I_{2h}^{(2)} z_h - z_h,$$

where $I_h^{(1)}$ and $I_{2h}^{(2)}$ denote the operators of cell-wise bilinear and patch-wise biquadratic interpolation, respectively (see Fig. 6.2). This results in the approximate error estimator

$$\begin{aligned} \big|J(e)\big| &\approx \sum_{K \in \mathbb{T}_h} \tilde{\eta}_K, \\ \tilde{\eta}_K &:= \left| \big(R(u_h), I_{2h}^{(2)} z_h - z_h\big)_K + \big(r(u_h), I_{2h}^{(2)} z_h - z_h\big)_{\partial K} \right|. \end{aligned} \tag{6.12}$$

This is to be compared with the traditional global "energy-norm" error estimator

$$\big\|\nabla(u - u_h)\big\| \le \eta_E := c_I c_S \left(\sum_{K \in \mathbb{T}_h} h_K^2 \rho_K(u_h)^2 \right)^{1/2} \tag{6.13}$$

with the cell residuals

$$\rho_K(u_h) := \big(\big\| R(u_h) \big\|_K^2 + \tfrac{1}{2} \big\| r(u_h) \big\|_{\partial K}^2 \big)^{1/2}$$

and certain interpolation and stability constants $c_I \approx 1$ and $c_S \approx 1$.

6.2.1 Balancing of Iteration and Discretization Error

In practice, the "exact" discrete solution $u_h \in V_h$ on the current mesh $\mathbb{T}_h$ is not known but rather an approximation $\tilde{u}_h \in V_h$ obtained by an iterative process $u_h^k \to u_h$ $(k \to \infty)$, such as a simple fixed point method (Gauß-Seidel), a Krylov space method (PCG), or a multigrid method (MG). Hence, in the a posteriori error representation

$$J(e) = \eta := \rho(u_h)(z - \psi_h),$$

we have to use this approximation $\tilde{u}_h := u_h^k$,

$$J(\tilde{e}) \approx \tilde{\eta} := \rho(\tilde{u}_h)(z - \psi_h) + ?$$

We need to balance the "iteration error" and the "discretization error" in order to have a useful stopping criterion (or fine tuning) for the iteration. Suppose that the adaptation process has generated a successively refined sequence of meshes $\mathbb{T}_l := \mathbb{T}_{h_l}$, $l = 0, \ldots, L$, and corresponding approximate discrete solutions $u_l \in V_l := V_{h_l}$.

Algorithm 6.1 Multigrid iteration MG($\mathbf{l}, \gamma, \mathbf{u_l^k}, \mathbf{f_l}$)

if $l = 0$ **then**
  Solve $A_0 u_0^{k+1} = f_0$ exactly.
else
  Pre-smoothing: $\bar{u}_l^k := S_l^{\nu}(u_l^k)$
  Residual: $d_l^k := f_l - A_l \bar{u}_l^k$
  Restriction: $\tilde{d}_{l-1}^k := r_l^{l-1} d_l^k$ (L^2 projection)
  for $r = 1$ **to** γ **do**
    Starting with $v_{l-1}^0 := 0$ iterate $v_{l-1}^r := \mathrm{MG}(l-1, \gamma, v_{l-1}^{r-1}, \tilde{d}_{l-1}^k)$
  end for
  Correction: $\bar{\bar{u}}_l^k := \bar{u}_l^k + p_{l-1}^l \tilde{v}_{l-1}^{\gamma}$ (natural embedding)
  Post-smoothing: $u_l^{k+1} := S_l^{\mu}(\bar{\bar{u}}_l^k)$
end if

Theorem 6.1 *Let $\tilde{u}_L, \tilde{z}_L \in V_L$ be any approximations to the exact primal and dual discrete solutions $u_L, z_L \in V_L$, respectively, on the finest mesh $\mathbb{T}_L$. Then, there holds the error representation*

$$J(u - \tilde{u}_L) = \rho(\tilde{u}_L)(z - \hat{z}_L) + \rho(\tilde{u}_L)(\hat{z}_L). \tag{6.14}$$

If a MG method has been used with canonical components, the following refined representation holds:

$$\rho(\tilde{u}_L)(\hat{z}_L) = \sum_{l=1}^{L} \big(R_l(\tilde{v}_l), \hat{z}_l - \hat{z}_{l-1}\big). \tag{6.15}$$

Here, $\hat{z}_l \in V_l$, $l = 0, \ldots, L$, can be chosen arbitrarily and $R_l(\tilde{v}_l)$ are the iteration residuals on the mesh levels $l = 0, \ldots, L$.

Proof [17] For the error $\tilde{e}_L := u - \tilde{u}_L$ there holds

$$\begin{aligned} J(\tilde{e}_L) &= a(\tilde{e}_L, z) = a(\tilde{e}_L, z - \hat{z}_L) + a(\tilde{e}_L, \hat{z}_L) \\ &= (f, z - \hat{z}_L) - a(\tilde{u}_L, z - \hat{z}_L) + (f, \hat{z}_L) - a(\tilde{u}_L, \hat{z}_L) \\ &= \rho(\tilde{u}_L)(z - \hat{z}_L) + \rho(\tilde{u}_L)(\hat{z}_L). \end{aligned}$$

If the multigrid method has been used, then the second term corresponding to the iteration error can be rewritten in the form

$$\rho(\tilde{u}_L)(\hat{z}_L) = \sum_{l=1}^{L} \big\{ (f, \hat{z}_l - \hat{z}_{l-1}) - a(\tilde{u}_L, \hat{z}_l - \hat{z}_{l-1}) \big\} + \big\{ (f, \hat{z}_0) - a(\tilde{u}_L, \hat{z}_0) \big\}.$$

Since $V_l \subset V_L$ for $l \leq L$, we observe by the definitions of Q_l (Ritz projection), P_l (L^2 projection), and A_l (discrete Laplacian) that for $\phi_l \in V_l$ there holds

$$(f, \phi_l) - a(\tilde{u}_L, \phi_l) = (P_l f, \phi_l) - (A_l Q_l \tilde{u}_L, \phi_l).$$

Further, by the identity $A_l Q_l = P_l A_L$ for $l \leq L$, we have

$$(P_l f, \phi_l) - (A_l Q_l \tilde{u}_L, \phi_l) = \big(P_l(f - A_L \tilde{u}_L), \phi_l\big) = \big(R_l(\tilde{u}_L), \phi_l\big).$$

Using the particular structure of the multigrid method, there holds

$$\begin{aligned} R_l(\tilde{u}_L) &= P_l(f_L - A_L \tilde{u}_L) \\ &= P_l f_L - P_l A_L S_L^{\nu}\big(\tilde{u}_L^{(0)}\big) - P_l A_L p_{L-1}^{L} \tilde{v}_{L-1} \\ &= P_l(d_L - A_{L-1} \tilde{v}_{L-1}) \\ &= P_l d_L - P_l A_{L-1} S_{L-1}^{\nu}\big(\tilde{v}_{L-1}^{(0)}\big) - P_l A_{L-1} p_{L-2}^{L-1} \tilde{v}_{L-2} \\ &\;\;\vdots \\ &= P_l(d_{l+2} - A_l \tilde{v}_{l+1}) \\ &= P_l d_{l+2} - P_l A_{l+1} S_{l+1}^{\nu}\big(\tilde{v}_{l+1}^{(0)}\big) - P_l A_{l+1} p_l^{l+1} \tilde{v}_l \\ &= P_l(d_{l+1} - A_l \tilde{v}_l) = R_l(\tilde{v}_l). \end{aligned}$$

Using this for $\phi_l = \hat{z}_l - \hat{z}_{l-1}$ and $\phi_0 = \hat{z}_0$ completes the proof. □

On the basis of the error representation (6.14), we use the following error balancing criterion:

$$\big|\rho(\tilde{u}_L)(\hat{z}_L)\big| \ll \big|\rho(\tilde{u}_L)(z - \hat{z}_L)\big|. \tag{6.16}$$

Since $\rho(u_L)(\hat{z}_L) = 0$ the term on the left tends to zero for proceeding iteration while the term on the right approaches the (generally) non-zero discretization error. Therefore, the left-hand term can be interpreted as measuring deviation from Galerkin

orthogonality of $\tilde{u}_L$ and the right-hand term is used for estimating the discretization error, however evaluated at the approximative solution $\tilde{u}_L$, i.e.,

$$\big|J(u-u_L)\big| \approx \rho(\tilde{u}_L)(z-\hat{z}_L), \qquad \big|J(u_L-\tilde{u}_L)\big| \approx \rho(\tilde{u}_L)(\hat{z}_L). \tag{6.17}$$

This heuristic concept is supported by the results of the test calculations presented below. It seems to be valid even on coarser meshes provided that the algebraic iteration is organized in a nested fashion, i.e., the approximate solution on the mesh $\mathbb{T}_{l-1}$ is used as the starting value for the iteration on the next refined mesh $\mathbb{T}_l$.

Remark 6.1 It is worth noting that:

1. The proof of the analogue of Theorem 6.1 for "energy-norm" and L^2-norm error control is due to [6].
2. The first error representation,

$$J(\tilde{e}_L) = \rho(\tilde{u}_L)(z-\hat{z}_L) + \rho(\tilde{u}_L)(\hat{z}_L),$$

   can be used for approximative solutions $\tilde{u}_L$ obtained by any solution process in V_L, such as simple fixed point iterations, Krylov space methods, or multigrid methods as well as perturbations caused by numerical quadrature or other "variational crimes".
3. The second error representation holds for V-, W-, or F-cycles and for any type of smoothing. It allows not only balancing the iteration against the discretization error but also tuning the smoothing iteration separately on the different mesh levels,

$$J(\tilde{e}_L) = \rho(\tilde{u}_L)(z-\hat{z}_L) + \sum_{l=1}^{L}\big(R_l(\tilde{v}_l), \hat{z}_l - \hat{z}_{l-1}\big).$$

   The corresponding adaptive algorithm is formulated below.

Algorithm 6.2 Adaptive algorithm

Choose an initial discretization $\mathbb{T}_{h_0}$ and set $l = 0$.
loop
  Set $k = 1$
  repeat
    if $k = 1$ **then**
      for $j = 0$ to l **do**
        Set $\nu_j = 1$, $\mu_j = 1$.
      end for
    end if
    Apply one multigrid cycle to the problem $A_l u_l = f_l$.
    Set $k = k + 1$.
    Evaluate the estimators η_{m_l} and η_{h_l}.

According to the error indicators on the different levels, $(R_j(\tilde{v}_j), \tilde{z}_j - p^j_{j-1}\tilde{z}_{j-1})$ determine the subset of levels $I = \{i_1, \ldots, i_n\}$ with the biggest contribution to the error estimator and increase the number of smoothing steps by
if $k > 1$ **then**
for $j = 1$ to n **do**
Set $\nu_{i_j} = 4$, $\mu_{i_j} = 4$.
end for
end if
until $|\eta_{m_l}| \leq c|\eta_{h_l}|$
if $|\eta_{h_l} + \eta_{m_l}| \leq \mathrm{TOL}$ **then**
stop
end if
Refine the mesh $\mathbb{T}_{h_l} \to \mathbb{T}_{h_{l+1}}$ accordingly to size of $\eta_{h_l,i}$.
Interpolate the previous solution $\tilde{u}_l$ on the mesh $\mathbb{T}_{h_{l+1}}$.
Increment l.
end loop

6.2.2 Numerical Tests

We consider a model Poisson problem (6.6) on a L-shaped domain $\Omega \subset \mathbb{R}^2$. The target value is the function value $J(u) := u(a)$ where $a = (0.2, 0.2)$. This irregular functional is regularized by

$$J_\varepsilon(u) := \big|B_\varepsilon(a)\big|^{-1} \int_{B_\varepsilon(a)} u(x)\,dx = u(a) + O\big(\varepsilon^2\big).$$

The discrete problems are solved by an MG method using a V-cycle and $4 + 4$ ILU-smoothing steps. The tolerance is $\mathrm{TOL} = 5 \times 10^{-7}$. By "MG I", we indicate iteration towards a round-off error level, while "MG II" refers to the use of an adaptive stopping criterion. The computational results are shown in Figs. 6.3, 6.4, and 6.5 as well as Tables 6.1 and 6.2. The "effectivity indices" for measuring the quality of the a posteriori error estimators are defined by

$$I^{\mathrm{tot}}_{\mathrm{eff}} := \frac{|J(e)|}{\eta_h + \eta_{\mathrm{it}}}, \qquad I^{h}_{\mathrm{eff}} := \frac{|J(e_h)|}{\eta_h}, \qquad I^{\mathrm{it}}_{\mathrm{eff}} := \frac{|J(e_{\mathrm{it}})|}{\eta_{\mathrm{it}}}.$$

Next, we consider the computation of the approximate solution u_h on a fixed locally refined, but still rather coarse, mesh by the Gauß-Seidel and the conjugate gradient (CG) method. The computational results are shown in Tables 6.3 and 6.4. In all cases the adaptive strategies proposed lead to significant work savings. Furthermore, the effectivity indices are close to one on finer meshes, which confirms the quality of the error estimators.

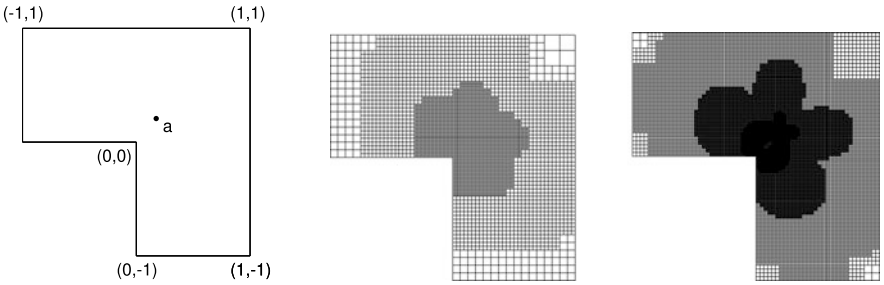


Fig. 6.3 Configuration and locally refined meshes

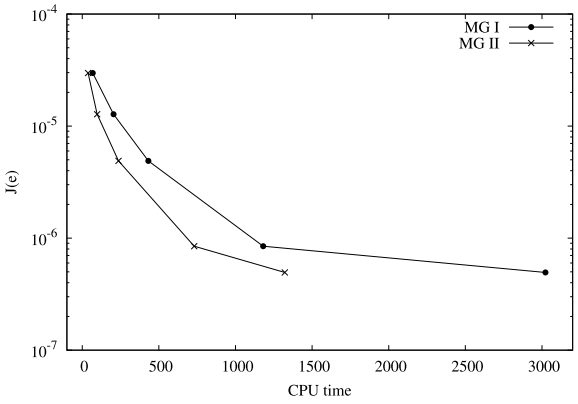


Fig. 6.4 Comparison of the CPU time used by the different MG methods MG I and MG II

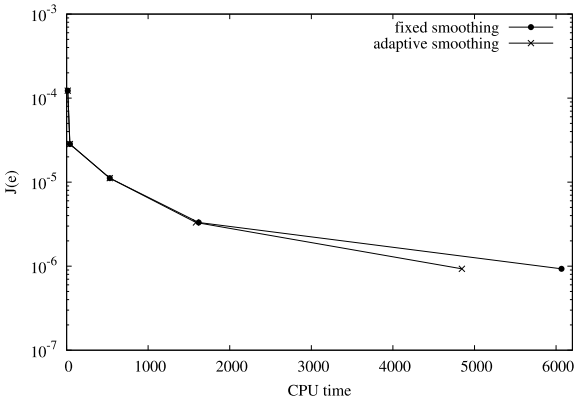


Fig. 6.5 Gain in efficiency of the multigrid algorithm by the adaptive choice of smoothing type and number of steps on the different mesh levels: $1+1$ ILU steps or $4+4$ ILU steps

Table 6.1 Iteration with *MG I* (iteration towards a round-off error level)

N	# Iter	$J(e)$	$\eta_h + \eta_{\text{it}}$	η_h	η_{it}	$I_{\text{eff}}^{\text{tot}}$
225	5	4.06e-03	1.57e-03	1.57e-03	6.01e-14	2.56
721	6	1.16e-03	9.57e-04	9.57e-04	3.95e-14	1.21
1 625	7	4.35e-04	2.26e-04	2.26e-04	4.70e-14	1.92
4 573	8	1.43e-04	9.95e-05	9.95e-05	7.71e-13	1.43
11 565	8	5.50e-05	2.98e-05	2.98e-05	1.67e-12	1.85
31 077	10	1.85e-05	1.28e-05	1.28e-05	6.33e-13	1.43
67 669	9	5.94e-06	4.89e-06	4.89e-06	2.67e-12	1.22
174 585	10	8.47e-07	2.00e-06	2.00e-06	1.79e-12	2.38
427 185	10	4.94e-07	7.63e-07	7.63e-07	1.37e-12	0.64

Table 6.2 Iteration with *MG II* (an adaptive stopping criterion)

N	# Iter	$J(e)$	$\eta_h + \eta_{\text{it}}$	η_h	η_{it}	$I_{\text{eff}}^{\text{tot}}$
225	1	4.06e-03	1.67e-03	1.58e-03	9.42e-05	2.44
721	2	1.16e-03	9.58e-04	9.57e-04	1.35e-06	1.21
1 625	1	4.35e-04	2.44e-04	2.26e-04	1.89e-05	1.19
4 573	2	1.43e-04	1.01e-04	9.95e-05	1.28e-06	1.43
11 565	2	5.50e-05	3.04e-05	2.98e-05	6.43e-07	1.82
31 077	2	1.85e-05	1.40e-05	1.28e-05	1.23e-06	1.32
67 669	2	5.94e-06	5.36e-06	4.89e-06	4.71e-07	1.11
174 585	3	8.47e-07	2.05e-06	2.00e-06	5.04e-08	0.41
427 185	3	4.94e-07	8.04e-07	7.63e-07	4.07e-08	0.64

Table 6.3 Gauss-Seidel iteration on a locally refined mesh with 721 knots (starting value taken from the preceding mesh)

Iter	$J(e_h)$	η_h	I_{eff}^h	$J(e_{\text{it}})$	η_{it}	$I_{\text{eff}}^{\text{it}}$	$\|u_L^{(k)} - u_L\|_\infty$
10	1.16e-3	9.42e-4	1.24	1.68e-3	1.65e-3	1.02	4.21e-2
20	1.16e-3	9.48e-4	1.22	1.21e-3	1.20e-3	1.01	3.66e-2
30	1.16e-3	9.51e-4	1.22	9.10e-4	9.01e-4	1.01	3.20e-2
40	1.16e-3	9.53e-4	1.22	6.86e-4	6.81e-4	1.01	2.78e-2
50	1.16e-3	9.54e-4	1.22	5.18e-4	5.15e-4	1.01	2.42e-2
60	1.16e-3	9.55e-4	1.22	3.90e-4	3.88e-4	1.00	2.10e-2
70	1.16e-3	9.55e-4	1.22	2.94e-4	2.93e-4	1.00	1.83e-2
80	1.16e-3	9.56e-4	1.22	2.21e-4	2.21e-4	1.00	1.59e-2
90	1.16e-3	9.56e-4	1.22	1.67e-4	1.66e-4	1.00	1.38e-2
100	1.16e-3	9.56e-4	1.22	1.25e-4	1.25e-4	1.00	1.19e-2

Table 6.4 CG iteration on a locally refined mesh with 721 knots (the starting value taken from the preceding mesh)

Iter	$J(e_h)$	η_h	I_{eff}^h	$J(e_{\text{it}})$	η_{it}	$I_{\text{eff}}^{\text{it}}$	$\|b - Ax^{(k)}\|_{A^{-1}}$
5	1.16e-3	9.50e-4	1.24	1.85e-03	1.80e-03	1.03	7.57e-3
10	1.16e-3	9.54e-4	1.22	4.60e-04	4.50e-04	1.03	6.34e-3
15	1.16e-3	9.50e-4	1.24	3.10e-05	2.99e-05	1.04	1.17e-3
20	1.16e-3	9.55e-4	1.22	2.17e-05	2.17e-05	1.01	3.08e-4
25	1.16e-3	9.57e-4	1.22	4.12e-06	4.12e-06	1.01	1.01e-4
30	1.16e-3	9.57e-4	1.22	1.09e-06	1.09e-06	1.00	1.32e-5
35	1.16e-3	9.57e-4	1.22	2.72e-07	2.72e-07	1.01	2.02e-6
40	1.16e-3	9.57e-4	1.22	8.22e-09	8.22e-09	1.00	2.31e-7
45	1.16e-03	9.57e-4	1.22	2.05e-09	2.05e-09	1.00	2.46e-08
50	1.16e-03	9.57e-4	1.22	1.93e-10	1.93e-10	1.00	1.94e-09

6.3 Eigenvalue Problems

Next, we consider the eigenvalue problem associated with the boundary value problem (6.6) of the Laplacian,

$$-\Delta u = \lambda u \quad \text{in } \Omega,\ u|_{\partial\Omega} = 0. \tag{6.18}$$

The corresponding variational formulation reads

$$a(u,\phi) = \lambda(u,\phi) \quad \forall \phi \in V = H_0^1(\Omega), \tag{6.19}$$

with the normalization $\|u\| = 1$. The corresponding Galerkin finite element approximation in $V_h \subset V$ reads

$$a(u_h,\phi_h) = \lambda_h(u_h,\phi_h) \quad \forall \phi_h \in V_h \tag{6.20}$$

with the normalization $\|u_h\| = 1$. The corresponding residual is given by

$$\begin{aligned}\rho(u_h,\lambda_h)(\psi) &:= \lambda_h(u_h,\psi) - a(u_h,\psi)\\ &= \sum_{K\in\mathbb{T}_h} \big\{ \big(R(u_h,\lambda_h),\psi\big)_K + \big(r(u_h),\psi\big)_{\partial K\setminus\partial\Omega} \big\},\end{aligned}$$

with the cell and edge residuals $R(u_h,\lambda_h)$ and $r(u_h)$ defined by

$$R(u_h)_{|K} := \lambda_h u_h + \Delta u_h, \qquad r(u_h)_{|\Gamma} := \begin{cases} -\frac{1}{2} n\cdot[\nabla u_h], & \text{if } \Gamma \subset \partial K \setminus \partial\Omega,\\ 0, & \text{if } \Gamma \subset \partial\Omega. \end{cases}$$

Theorem 6.2 *Let $\{\tilde{u}_h,\tilde{\lambda}_h\} \in V_h \times \mathbb{R}$, $\|\tilde{u}_h\| = 1$, be any approximation to the discrete eigenpair $\{u_h,\lambda_h\} \in V_h \times \mathbb{R}$ on the current mesh $\mathbb{T}_h$. Then, there holds*

$$(\tilde{\lambda}_h - \lambda)(1-\sigma_h) = \rho(\tilde{u}_h,\tilde{\lambda}_h)(u-\phi_h) + \rho(\tilde{u}_h,\tilde{\lambda}_h)(\phi_h) \tag{6.21}$$

for an arbitrary $\phi_h \in V_h$. *Here* $\sigma_h := \frac{1}{2}\|\tilde{u}_h - u\|^2$.

Proof [21] Observing $\|\tilde{u}_h\| = \|u\| = 1$, there holds

$$\begin{aligned}
&\rho(\tilde{u}_h, \tilde{\lambda}_h)(u - \phi_h) + \rho(\tilde{u}_h, \tilde{\lambda}_h)(\phi_h) \\
&\quad = \tilde{\lambda}_h(\tilde{u}_h, u) - a(\tilde{u}_h, u) \\
&\quad = (\tilde{\lambda}_h - \lambda)(\tilde{u}_h, u) + \lambda(\tilde{u}_h, u) - a(\tilde{u}_h, u) \\
&\quad = (\tilde{\lambda}_h - \lambda)(\tilde{u}_h, u) \\
&\quad = (\tilde{\lambda}_h - \lambda)\big(\tfrac{1}{2}\|\tilde{u}_h\|^2 + \tfrac{1}{2}\|u\|^2 - \tfrac{1}{2}\|\tilde{u}_h - u\|^2\big) \\
&\quad = (\tilde{\lambda}_h - \lambda)(1 - \sigma_h).
\end{aligned}$$

□

Remark 6.2 It is worth noting that:

1. The error representation has to be evaluated for a convergent sequence of approximate eigenfunctions: $\|\tilde{u}_h - u\|^2 \to 0$.
2. The evaluation of the error representation requires higher-order approximations $\hat{u}_h \approx u$ and $\hat{\sigma}_h \approx \sigma_h$ obtained, for example, from $\tilde{u}_h$ by post-processing as described above:

$$\tilde{\lambda}_h - \lambda \approx \frac{1}{1 - \hat{\sigma}_h}\big\{\rho(\tilde{u}_h, \tilde{\lambda}_h)(\hat{u}_h - \tilde{u}_h) + \rho(\tilde{u}_h, \tilde{\lambda}_h)(\tilde{u}_h)\big\}. \tag{6.22}$$

3. The second term on the right-hand side represents the deviation from Galerkin orthogonality and can be evaluated without any approximation.
4. The error representation (6.21) has a natural extension to non-symmetric eigenvalue problems (non-deficient eigenvalues):

$$\begin{aligned}
(\tilde{\lambda}_h - \lambda) \approx \frac{1}{1 - \hat{\sigma}_h}\big\{&\tfrac{1}{2}\rho(\tilde{u}_h, \tilde{\lambda}_h)\big(\hat{u}_h^* - \tilde{u}_h^*\big) + \tfrac{1}{2}\rho^*\big(\tilde{u}_h^*, \tilde{\lambda}_h\big)(\hat{u}_h - \tilde{u}_h) \\
&+ \tfrac{1}{2}\rho(\tilde{u}_h, \tilde{\lambda}_h)\big(\tilde{u}_h^*\big) + \tfrac{1}{2}\rho^*\big(\tilde{u}_h^*, \tilde{\lambda}_h\big)(\tilde{u}_h)\big\},
\end{aligned} \tag{6.23}$$

where $\hat{\sigma}_h := \frac{1}{2}(\tilde{u}_h - \hat{u}_h, \tilde{u}_h^* - \hat{u}_h^*)$, and $\tilde{u}_h^*$ is an approximation to the adjoint eigenfunction u^* corresponding to the eigenvalue λ. In the non-degenerate case, we can use the normalization $(u_h, u_h^*) = 1$ (see [15, 21]).

The results of various test computations reported in [21] demonstrate that our general approach to balancing discretization and iteration error also works well for symmetric as well as nonsymmetric eigenvalue problems. Based on the a posteriori error representations (6.21) or (6.23), we obtain effective stopping criteria for Krylov-space methods such as, for example, the Arnoldi method (see [22, 24]), which result in significant work savings.

6.4 Saddle Point Problems

The approach for simultaneous estimation of discretization and iteration errors introduced above can also be used for indefinite (linear) systems such as saddle point problems. We illustrate this for two different kinds of saddle point problems, the Stokes equation for modeling incompressible creeping viscous flow and the Karush-Kuhn-Tucker (KKT) system occurring as a first-order optimality condition of linear-quadratic optimal control problems.

6.4.1 Stokes Equations

The Stokes equation of fluid mechanics describes the behavior of a creeping incompressible fluid occupying a domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$,

$$\begin{aligned} &-\nu\Delta v + \nabla p = 0, \qquad \nabla \cdot v = 0 \quad \text{in } \Omega, \\ &v = 0 \quad \text{on } \Gamma_{\text{rigid}}, \qquad v = v^{\text{in}} \quad \text{on } \Gamma_{\text{in}}, \qquad \nu\partial_n v - pn = 0 \quad \text{on } \Gamma_{\text{out}}. \end{aligned} \tag{6.24}$$

The boundary is split like $\partial\Omega = \Gamma_{\text{rigid}} \cup \Gamma_{\text{in}} \cup \Gamma_{\text{out}}$, where Γ_{rigid} is the rigid part, Γ_{in} the inflow part, and Γ_{out} the usually artificial outflow part. For the meaning and properties of the Neumann-type outflow boundary condition (so-called "do-nothing" condition), we refer to [16]. Here, we consider the two-dimensional benchmark problem "channel flow around an obstacle" introduced in [23] (see Fig. 6.6). The quantity of interest is the drag coefficient

$$J(u) := \frac{2}{\bar{U}^2 D} \int_S n^T \big(2\nu\tau(v) - pI\big)e_1 \, \mathrm{ds},$$

where $u = \{v, p\}$, $\tau(v) := \frac{1}{2}(\nabla v + \nabla v^T)$ the strain tensor, n the outer normal unit vector along S, D the diameter of the obstacle, $\bar{U}$ the maximum inflow velocity, and e_1 the unit vector in the main flow direction. The variational formulation of the

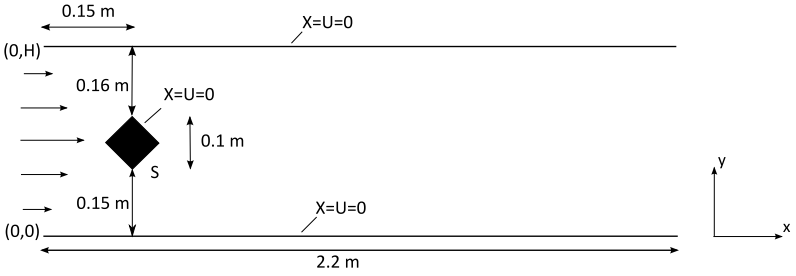


Fig. 6.6 Configuration of the flow example

Table 6.5 Iteration with *MG I* (iteration towards a round-off error level)

N	# Iter	$J(e)$	$\eta_h + \eta_{\text{it}}$	η_h	η_{it}	$I_{\text{eff}}^{\text{tot}}$
708	12	5.69e-05	9.19e-05	9.19e-05	2.03e-18	0.62
1 754	9	3.12e-05	2.81e-05	2.81e-05	1.05e-16	1.11
4 898	9	1.83e-05	1.21e-05	1.21e-05	2.20e-15	1.52
11 156	9	1.05e-05	7.01e-06	7.01e-06	9.49e-15	1.49
22 526	10	5.34e-06	3.77e-06	3.77e-06	8.36e-17	1.41
44 874	10	2.75e-06	2.12e-06	2.12e-06	3.39e-16	1.30
82 162	10	1.26e-06	1.09e-06	1.09e-06	4.29e-17	1.16
159 268	11	5.76e-07	6.11e-07	6.11e-07	1.26e-17	1.06
306 308	12	1.85e-07	2.98e-07	2.98e-07	8.74e-19	1.61

Table 6.6 Iteration with *MG II* (an adaptive stopping criterion)

N	# Iter	$J(e)$	$\eta_h + \eta_{\text{it}}$	η_h	η_{it}	$I_{\text{eff}}^{\text{tot}}$
708	2	5.69e-05	9.74e-05	9.17e-05	5.62e-06	0.59
1 754	2	3.12e-05	2.82e-05	2.81e-05	6.81e-08	1.11
4 898	2	1.83e-05	1.21e-05	1.21e-05	1.60e-08	1.52
11 156	2	1.05e-05	7.05e-06	7.01e-06	3.42e-08	1.49
22 526	2	5.34e-06	3.82e-06	3.77e-06	5.48e-08	1.39
44 874	2	2.75e-06	2.16e-06	2.12e-06	4.04e-08	1.28
82 162	2	1.27e-06	1.11e-06	1.09e-06	2.63e-08	1.14
159 268	2	5.76e-07	6.41e-07	6.10e-07	3.07e-08	0.90
306 308	2	1.86e-07	3.10e-07	2.97e-07	1.31e-08	0.60

problem (6.24) reads: Find $\{v, p\} \in (\hat{v}^{\text{in}} + H) \times L$ satisfying

$$\begin{aligned} \nu(\nabla v, \nabla \phi) - (p, \nabla \cdot \phi) &= (f, \phi) \quad \forall \phi \in H, \\ (\chi, \nabla \cdot v) &= 0 \qquad \forall \chi \in L, \end{aligned}$$

where $H := H_0^1(\Gamma_{\text{rigid}} \cup \Gamma_{\text{in}}; \Omega)^2$, $L := L^2(\Omega)$, and $\hat{v}^{\text{in}}$ is a suitable (solenoidal) extension of the boundary data.

The discretization uses equal-order (bilinear) Q_1 elements for velocity and pressure with additional pressure stabilization for circumventing the usual "inf-sup" stability condition,

$$\begin{aligned} \nu(\nabla v_h, \nabla \phi_h) - (p_h, \nabla \cdot \phi_h) &= (f, \phi_h) \quad \forall \phi_h \in H_h, \\ (\chi_h, \nabla \cdot v_h) + s_h(\chi_h, p_h) &= 0 \qquad \forall \chi_h \in L_h, \end{aligned} \tag{6.25}$$

where $V_h \subset V$ and $L_h \subset L$ are the finite element subspaces and $s_h(\chi_h, p_h)$ is a stabilizing form. For more details on pressure stabilization, we refer to the survey articles

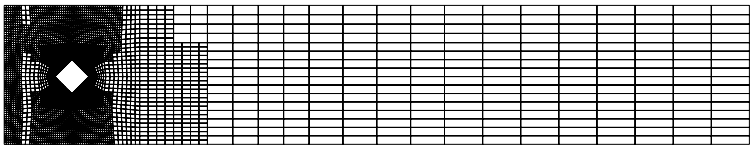


Fig. 6.7 A refined mesh with 4 898 knots in the flow example

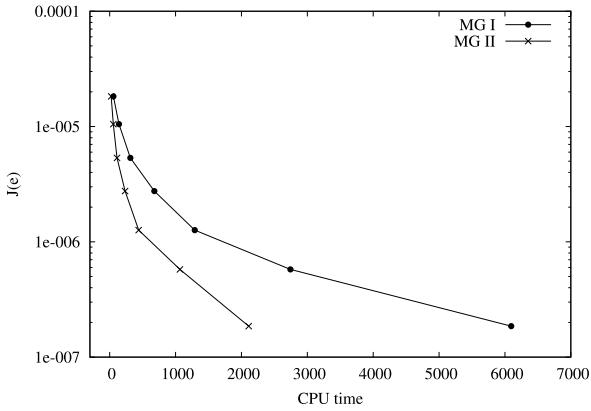


Fig. 6.8 Comparison of the CPU time used by the two *MG* variants *MG I* and *MG II*

[18, 19]. The discrete saddle point problem (6.25) is solved by an MG method using the canonical mesh transfer operations and "block ILU" smoothing (with $4+4$ smoothing steps). The computational results are shown in Tables 6.5 and 6.6 as well as Figs. 6.7 and 6.8.

6.4.2 The KKT System of Linear-Quadratic Optimization Problems

We consider the linear-quadratic optimization problem

$$\begin{aligned} J(u,q) &:= \tfrac{1}{2}\|u-\bar{u}\|^2 + \tfrac{1}{2}\alpha\|q\|^2 \longrightarrow \min, \\ -\Delta u &= f+q \quad \text{in } \Omega, \qquad u=0 \quad \text{on } \partial\Omega, \end{aligned} \tag{6.26}$$

on $\Omega := (0,1)^2 \subset \mathbb{R}^2$ with the force term f, prescribed target distribution $\bar{u}$, and distributed control q. The regularization parameter is taken as $\alpha = 10^{-3}$. This problem is solved by the Euler-Lagrange approach, which uses the Lagrangian functional

$$\mathscr{L}(u,q,\lambda) := J(u,q) + (f+q,\lambda) - (\nabla u, \nabla\lambda),$$

with the adjoint variable $\lambda \in V := H_0^1(\Omega)$. Then, for any optimal solution $\{u,q\} \in V \times Q := H_0^1(\Omega) \times L^2(\Omega)$ there exists an adjoint solution $\lambda \in V$ such that the

triplet $\{u, q, \lambda\} \in V \times Q \times V$ is a stationary point of the Lagrangian, i.e., it solves the following (linear) saddle point system:

$$\begin{aligned} (\nabla\phi, \nabla\lambda) - (u, \phi) &= -(\bar{u}, \phi) && \forall \phi \in V, \\ (\chi, \lambda) + \alpha(\chi, q) &= 0 && \forall \chi \in Q, \\ (\nabla u, \nabla\psi) - (q, \psi) &= (f, \psi) && \forall \psi \in V. \end{aligned} \tag{6.27}$$

This first-order necessary optimality condition is the so-called Karush-Kuhn-Tucker (KKT) system of the optimization problem.

For solving the KKT system (6.27), we use conforming bilinear Q_1 elements for all three variables $\{u, q, \lambda\}$. Denoting the corresponding finite element subspaces by $V_h \subset V$ and $Q_h \subset Q$, we obtain the discrete saddle point problem

$$\begin{aligned} (\nabla\phi_h, \nabla\lambda_h) - (u_h, \phi_h) &= -(\bar{u}, \phi_h) && \forall \phi_h \in V_h, \\ (\chi_h, \lambda_h) + \alpha(\chi_h, q_h) &= 0 && \forall \chi_h \in Q_h, \\ (\nabla u_h, \nabla\psi_h) - (q_h, \psi_h) &= (f, \psi_h) && \forall \psi_h \in V_h. \end{aligned} \tag{6.28}$$

This reads in a strong form as

$$\begin{aligned} -\Delta\lambda - u &= -\bar{u}, && \text{in } \Omega,\ \lambda|_{\partial\Omega} = 0, \\ \lambda + \alpha q &= 0, && \text{in } \Omega, \\ -\Delta u - q &= f, && \text{in } \Omega,\ u|_{\partial\Omega} = 0. \end{aligned} \tag{6.29}$$

This linear algebraic saddle point problem is again solved by a MG method using a block ILU iteration as a smoother.

Theorem 6.3 *Let $\{u, q, \lambda\} \in V \times Q \times V$ be the solution of the KKT system and $\{\tilde{u}_h, \tilde{q}_h, \tilde{\lambda}_h\} \in V_h \times Q_h \times V_h$ the approximative finite element solution of the discrete KKT system on the current mesh $\mathbb{T}_h$. Then, we have the error representation*

$$\begin{aligned} J(u, q) - J(\tilde{u}_h, \tilde{q}_h) = \tfrac{1}{2}\rho^*(\tilde{u}_h, \tilde{\lambda}_h)(u - \tilde{u}_h) + \tfrac{1}{2}\rho^q(\tilde{q}_h, \tilde{\lambda}_h)(q - \tilde{q}_h) \\ + \tfrac{1}{2}\rho(\tilde{u}_h, \tilde{q}_h)(\lambda - \tilde{\lambda}_h) + \rho(\tilde{u}_h, \tilde{q}_h)(\tilde{\lambda}_h), \end{aligned} \tag{6.30}$$

with the residuals

$$\begin{aligned} \rho^*(\tilde{u}_h, \tilde{\lambda}_h)(\phi) &:= (\tilde{u}_h - \bar{u}, \phi) - (\nabla\phi, \nabla\tilde{\lambda}_h), \\ \rho^q(\tilde{q}_h, \tilde{\lambda}_h)(\phi) &:= \alpha(\phi, \tilde{q}_h) + (\phi, \tilde{\lambda}_h), \\ \rho(\tilde{u}_h, \tilde{q}_h)(\phi) &:= (f + \tilde{q}_h, \phi) - (\nabla\tilde{u}_h, \nabla\phi). \end{aligned}$$

Proof For the proof, we refer to [17]. □

Remark 6.3 The choice of the cost functional $J(\cdot, \cdot)$ for error control may not be considered appropriate in the present case of a tracking problem where the particular

Table 6.7 *MG II* with block ILU smoothing, $\alpha = 10^{-3}$

N	E_{tot}	# Iter	E_h	η_h	I^h_{eff}	E_{it}	η_{it}	$I^{\text{it}}_{\text{eff}}$
25	9.35e-4	2	9.35e-4	1.83e-3	0.51	1.14e-07	1.97e-07	0.58
81	1.64e-4	2	1.78e-4	2.19e-4	0.82	1.42e-05	1.68e-05	0.85
289	3.75e-5	2	4.16e-5	4.39e-5	0.95	4.13e-06	4.33e-06	0.96
1 089	1.05e-5	2	1.02e-5	1.03e-5	0.99	3.48e-07	3.52e-07	0.99
3 985	2.67e-6	2	2.54e-6	2.55e-6	1.00	1.28e-07	1.28e-07	1.00
13 321	6.65e-7	2	6.48e-7	6.49e-7	1.00	1.63e-08	1.63e-08	1.00
47 201	1.76e-7	2	1.70e-7	1.69e-7	1.01	6.76e-09	6.77e-09	1.00
163 361	4.89e-8	2	4.69e-8	4.68e-8	1.01	1.97e-09	1.97e-09	1.00
627 697	1.23e-8	2	1.21e-8	1.21e-8	1.01	2.13e-10	2.13e-10	1.00

least-squares form of the functional is somewhat arbitrary. Instead, one may want to measure the solution accuracy rather in terms of some more relevant quantity depending on control and state, such as for example the norm $\|q - \tilde{q}_h\|_Q$ of the error in the control. This can be accomplished by utilizing an additional "outer" dual problem such as described in [5, 9].

We consider the example with the target distribution

$$\bar{u} = \tfrac{2\pi^2-1}{2\pi^2} \sin(\pi x) \sin(\pi y)$$

and the exact solution

$$u = -\frac{1}{2\pi^2} \sin(\pi x) \sin(\pi y), \qquad q = \frac{1}{2\alpha\pi^2} \sin(\pi x) \sin(\pi y),$$

$$\lambda = -\frac{1}{2\pi^2} \sin(\pi x) \sin(\pi y).$$

The forcing term f is accordingly adjusted. For simplicity, the discrete state and control spaces are chosen the same, $V_h = Q_h$, using isoparametric bilinear shape functions. For this test, we use the MG II algorithm with the stopping criterion

$$\eta_{\text{it}} \le \frac{1}{10}\, \eta_h.$$

First, we solve the discretized KKT system by the adaptive multigrid method using the V-cycle and again $4 + 4$-block-ILU smoothing steps on each level. Then, we use the multigrid method with only one undamped block-Jacobi smoothing step. The results are shown in Tables 6.7 and 6.8, where we use the abbreviations

$$E_{\text{tot}} := \big|J(u,q) - J(\tilde{u}_h, \tilde{q}_h)\big|, \qquad E_h := \big|J(u,q) - J(u_h, q_h)\big|,$$
$$E_{\text{it}} := \big|J(u_h,q_h) - J(\tilde{u}_h, \tilde{q}_h)\big|.$$

Table 6.8 *MG II* with block Jacobi smoothing, $\alpha = 10^{-3}$

N	E_{tot}	# Iter	E_h	η_h	I^h_{eff}	E_{it}	η_{it}	$I^{\text{it}}_{\text{eff}}$
25	9.44e-4	4	1.83e-3	9.35e-4	1.96	1.55e-5	8.99e-6	1.73
81	1.84e-4	5	2.20e-4	1.78e-4	1.23	7.59e-6	6.44e-6	1.18
289	4.36e-5	5	4.40e-5	4.16e-5	1.05	2.04e-6	1.96e-6	1.04
1 089	1.10e-5	4	1.03e-5	1.02e-5	1.01	8.53e-7	8.44e-7	1.01
3 985	2.69e-6	4	2.55e-6	2.56e-6	0.99	1.31e-7	1.30e-7	1.00
13 321	6.94e-7	4	6.47e-7	6.69e-7	0.96	2.51e-8	2.51e-8	1.00
47 201	1.95e-7	4	1.69e-7	1.90e-7	0.88	4.39e-9	4.40e-9	1.00
171 969	7.24e-8	3	4.42e-8	6.93e-8	0.63	3.07e-9	3.10e-9	0.99

We observe again a significant work saving by using the adaptive stopping criterion of the iteration.

6.5 The Nonlinear Case

Finally, we describe how our approach to the simultaneous estimation of the discretization and iteration errors extends to nonlinear variational problems of the form

$$A(u)(\psi) = F(\psi) \quad \forall \psi \in V, \; J(u) = ? \tag{6.31}$$

with a semi-linear "energy form" $A(\cdot)(\cdot)$ and a nonlinear output functional $J(\cdot)$ defined on the solution space V (both assumed to be sufficiently often differentiable). The starting point is the observation that any solution of the "primal" problem (6.31) corresponds to a stationary point of the Lagrangian functional $\mathscr{L}(u, z) := J(u) + F(z) - A(u)(z)$ with the dual variable $z \in V$ (Lagrangian multiplier). This results in the system

$$\begin{aligned} A(u)(\psi) &= F(\psi) && \forall \psi \in V, \\ A'(u)(\phi, z) &= J'(u)(\phi) && \forall \phi \in V. \end{aligned} \tag{6.32}$$

The finite element discretization of this system in spaces $V_h \subset V$ seeks primal and dual approximation $\{u_h, z_h\} \in V_h \times V_h$ satisfying

$$\begin{aligned} A(u_h)(\psi_h) &= F(\psi_h) && \forall \psi_h \in V_h, \\ A'(u_h)(\phi_h, z_h) &= J'(u_h)(\phi_h) && \forall \phi_h \in V_h. \end{aligned} \tag{6.33}$$

The corresponding primal and dual residuals are defined by

$$\rho(u_h)(\cdot) := F(\cdot) - A(u_h)(\cdot), \qquad \rho^*(u_h, z_h)(\cdot) := J'(u_h)(\cdot) - A'(u_h)(\cdot, z_h).$$

Theorem 6.4 *Let $\tilde{u}_h, \tilde{z}_h \in V_h$ be any approximations to the primal and dual discrete solutions $u_h, z_h \in V_h$ on the current mesh $\mathbb{T}_h$. Then, there holds*

$$J(u) - J(\tilde{u}_h) = \tfrac{1}{2}\rho(\tilde{u}_h)(z - \tilde{z}_h) + \tfrac{1}{2}\rho^*(\tilde{u}_h, \tilde{z}_h)(u - \tilde{u}_h) + \rho(\tilde{u}_h)(\tilde{z}_h) + \tilde{R}_h^{(3)} \quad (6.34)$$

with a remainder $\tilde{R}_h^{(3)}$ cubic in the errors $u - \tilde{u}_h$ and $z - \tilde{z}_h$.

Proof [20] For pairs $x = \{u, z\}$, we set $L(x) := \mathscr{L}(u, z)$. Then, with the abbreviation $\tilde{e}^z := u - \tilde{u}_h$, $\tilde{e}^z := z - \tilde{z}_h$, and $\tilde{e} := \{\tilde{e}^u, \tilde{e}^z\}$, there holds

$$\begin{aligned} J(u) - J(\tilde{u}_h) &= L(x) \underbrace{-F(z) + A(u)(z)}_{=0} - L(\tilde{x}_h) + \underbrace{F(\tilde{z}_h) - A(\tilde{u}_h)(\tilde{z}_h)}_{\neq 0} \\ &= \int_0^1 L'(\tilde{x}_h + s\tilde{e})(\tilde{e})\, ds + F(\tilde{z}_h) - A(\tilde{u}_h)(\tilde{z}_h). \end{aligned}$$

For the integral, we use the trapezoidal rule with integral remainder as follows:

$$\begin{aligned} J(u) - J(\tilde{u}_h) &= \tfrac{1}{2}\big\{\underbrace{L'(x)(\tilde{e})}_{=0} + \mathscr{L}'(\tilde{x}_h)(\tilde{e})\big\} \\ &\quad + \underbrace{\tfrac{1}{2}\int_0^1 L'''(\tilde{x}_h + s\tilde{e})(\tilde{e}, \tilde{e}, \tilde{e}) s(s-1)\, ds}_{=: \tilde{R}_h^{(3)}} + \underbrace{F(\tilde{z}_h) - A(\tilde{u}_h)(\tilde{z}_h)}_{= \rho(\tilde{u}_h)(\tilde{z}_h)} \\ &= \tfrac{1}{2}L'(\tilde{x}_h)(\tilde{e}) + \tilde{R}_h^{(3)} + \rho(\tilde{u}_h)(\tilde{z}_h) \\ &= \tfrac{1}{2}\big\{F(\tilde{e}^z) - A(\tilde{u}_h)(\tilde{e}^z) + J'(\tilde{u}_h)(\tilde{e}^u) - A'(\tilde{u}_h)(\tilde{e}^u, \tilde{z}_h)\big\} \\ &\quad + \tilde{R}_h^{(3)} + \rho(\tilde{u}_h)(\tilde{z}_h) \\ &= \tfrac{1}{2}\rho(\tilde{u}_h)(z - \tilde{z}_h) + \tfrac{1}{2}\rho^*(\tilde{u}_h, \tilde{z}_h)(u - \tilde{u}_h) + \tilde{R}_h^{(3)} + \rho(\tilde{u}_h)(\tilde{z}_h). \quad \square \end{aligned}$$

Remark 6.4 We make the following remarks:

1. The cubic remainder term $\tilde{R}_h^{(3)}$ is neglected or monitored by replacing $u - u_h^k \approx u_h^{k+1} - u_h^k$ and $z - z_h^k \approx z_h^{k+1} - z_h^k$.
2. For non-unique solutions the following a priori assumption $\{u_h, z_h\} \to \{u, z\}$ for $h \to 0$ is needed.
3. We have to solve the *linear* discrete dual problem

$$A'(u_h)(\phi_h, z_h) = J'(u_h)(\phi_h) \quad \forall \phi_h \in V_h. \quad (6.35)$$

4. The weights in the error representation are again approximated by patch-wise higher-order interpolation: $(z - \tilde{z}_h)_{|K} \approx (\tilde{I}_{2h}^{(2)}\tilde{z}_h - \tilde{z}_h)_{|K}$. The steps 1–4 are the essence of the **D**ual **W**eighted **R**esidual (DWR) method applied to the Galerkin finite element approximation of nonlinear problems.

5. The error representation (6.34) can be used to control the accuracy in the Newton iteration or in any other simple fixed point iteration for solving the algebraic problem (6.32).
6. If the approximative discrete solution $\tilde{u}_h$ is obtained by the Newton method, also an adaptive stopping criterion is needed for the inner linear solver of the single Newton steps. Such a strategy for simultaneous control of a discretization error, an outer *nonlinear* iteration error, and an inner *linear* iteration error can be developed on the basis of an a posteriori error representation by exploiting the structure of the Newton methods. For details, we refer to the forthcoming paper [20].

6.5.1 Numerical Example

We consider the following simple test problem: Compute $J(u) := u_1(a)$ for the solution $u \in V := H_0^1(\Omega)^2$ of the nonlinear system

$$\begin{aligned} -\Delta u_1 + 2u_2^2 &= 1, & u_1|_{\partial\Omega} &= 0, \\ -\Delta u_2 + u_1 u_2 &= 0, & u_2|_{\partial\Omega} &= 0. \end{aligned} \tag{6.36}$$

The configuration is shown in Fig. 6.9.

In this case the corresponding variational formulation reads

$$\begin{aligned} A(u)(\phi) &:= (\nabla u_1, \nabla \phi_1) + 2\big(u_1^2, \phi_1\big) + (\nabla u_2, \nabla \phi_2) + (u_1 u_2, \phi_2) \\ &= F(\phi) := (f, \phi) \quad \forall \phi \in V. \end{aligned} \tag{6.37}$$

For the discretization of the problem (6.37), we use again a standard finite element method with continuous Q_1 elements. The resulting nonlinear algebraic problems are solved by a damped Newton method with damping a factor $\theta = 0.5$,

$$A'\big(u_h^t\big)\big(u_h^{t+1}, \phi_h\big) = A'\big(u_h^t\big)\big(u_h^t, \phi_h\big) - \theta\big\{F(\phi_h) - A\big(u_h^t\big)(\phi_h)\big\}, \quad \forall \phi_h \in V_h. \tag{6.38}$$

We consider the following two different stopping criteria:

- *Newton I:* Reduction of initial Newton residual by factor 10^{-11};
- *Newton II:* Iteration error $\approx 10^{-1} \times$ discretization error.

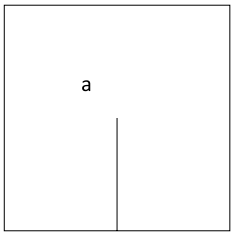


Fig. 6.9 Configuration of the nonlinear test problem: slit domain and point value evaluation

Table 6.9 *Newton I:* Iteration towards a "round-off error level" 10^{-11}

N	# Iter	$J(e)$	$\eta_h + \eta_{\text{it}}$	η_h	η_{it}	$I_{\text{eff}}^{\text{tot}}$
85	31	3.31e-03	1.49e-03	1.49e-03	1.69e-11	2.22
297	29	1.24e-03	5.78e-04	5.78e-04	7.14e-11	2.13
897	29	5.46e-04	2.30e-04	2.30e-04	7.26e-11	2.38
2063	29	2.43e-04	9.59e-05	9.59e-05	7.32e-11	2.56
4537	27	1.14e-04	4.34e-05	4.34e-05	2.94e-10	2.63
9969	27	5.28e-05	2.15e-05	2.15e-05	2.94e-10	2.44
21389	27	2.23e-05	1.03e-05	1.03e-05	2.94e-10	2.17
39549	27	7.58e-06	5.36e-06	5.36e-06	2.94e-10	1.41

Table 6.10 *Newton II:* An adaptive stopping criterion

N	# Iter	$J(e)$	$\eta_h + \eta_{\text{it}}$	η_h	η_{it}	$I_{\text{eff}}^{\text{tot}}$
85	8	3.31e-03	1.63e-03	1.49e-03	1.41e-04	2.13
297	10	1.24e-03	6.15e-04	5.77e-04	3.74e-05	2.08
897	11	5.46e-04	2.49e-04	2.30e-04	1.90e-05	2.27
2063	13	2.43e-04	1.01e-04	9.59e-05	4.79e-06	2.44
4537	14	1.14e-04	4.58e-05	4.34e-05	2.40e-06	2.56
9969	15	5.28e-05	2.27e-05	2.15e-05	1.20e-06	2.38
21389	16	2.23e-05	1.09e-05	1.03e-05	6.03e-07	2.08
39549	17	7.58e-06	5.66e-06	5.36e-06	3.01e-07	1.39

The linear subproblems are solved by an MG iteration with

- Smoother: Jacobi with damping factor 0.5;
- Stopping criterion: Reduction of the initial multigrid residual by factor 10^{-11}.

The obtained results are shown in Tables 6.9 and 6.10. Again, we observe significant work savings through the adaptive stopping criterion. The effectivity indices are relatively close to one, even on coarser meshes, which demonstrates the sharpness of our error indicators. However, we observe slight underestimation on all meshes.

6.6 Conclusion and Outlook

Goal-oriented adaptivity by the DWR method is in principle possible for all problems formulated within a variational setting. Though largely of heuristic nature the DWR method provides a general guideline for treating even most complex nonlinear systems. However, its theoretical justification in any particular case requires additional assumptions and hard work. In this way the discretization error and the algebraic iteration error, linear as well as nonlinear, can be simultaneously controlled

leading to effective stopping criteria and significant work savings. Current developments into the same direction are a posteriori control of the following additional "variational crimes":

- Quadrature error,
- Boundary approximation,
- Stabilization error ("inf-sup" and "transport" stabilization),
- Domain approximation (truncation of unbounded domains),
- Various modeling errors.

This will be the subject of forthcoming papers.

References

1. Babuška I, Strouboulis T (2001) The finite element method and its reliability. Clarendon Press, New York
2. Bangerth W, Rannacher R (2003) Adaptive finite element methods for differential equations. Birkhäuser, Basel
3. Bank RE, Dupont T (1981) An optimal order process for solving finite element equations. Math Comput 36(153):35–51
4. Becker R (1998) An adaptive finite element method for the Stokes equations including control of the iteration error. In: Enumath 97: 2nd European conference on numerical mathematics and advanced applications, Heidelberg, 1997. World Scientific, River Edge, pp 609–620
5. Becker R, Braack M, Meidner D, Rannacher R, Vexler B (2007) Adaptive finite element methods for pde-constrained optimal control problems. In: Jäger W, Rannacher R, Warnatz J (eds) Reactive flow, diffusion and transport. Springer, Berlin, pp 177–205
6. Becker R, Johnson C, Rannacher R (1995) Adaptive error control for multigrid finite element methods. Computing 55(4):271–288
7. Becker R, Rannacher R (1996) A feed-back approach to error control in finite element methods: basic analysis and examples. East-West J Numer Math 4(4):237–264
8. Becker R, Rannacher R (2001) An optimal control approach to a posteriori error estimation in finite element methods. Acta Numer 10:1–102
9. Becker R, Vexler B (2004) A posteriori error estimation for finite element discretization of parameter identification problems. Numer Math 96(3):435–459
10. Bramble JH (1993) Multigrid methods. Pitman research notes in mathematics, vol 294. Longman, Harlow
11. Bramble JH, Pasciak JE (1993) New estimates for multilevel algorithms including the V-cycle. Math Comput 60(202):447–471
12. Bramble JH, Pasciak JE, Wang JP, Xu J (1991) Convergence estimates for multigrid algorithms without regularity assumptions. Math Comput 57(195):23–45
13. Ciarlet PG (2002) The finite element method for elliptic problems. Classics appl math, vol 40. SIAM, Philadelphia
14. Hackbusch W (1985) Multigrid methods and applications. Springer, Berlin
15. Heuveline V, Rannacher R (2001) A posteriori error control for finite element approximations of elliptic eigenvalue problems. Adv Comput Math 15(1–4):107–138
16. Heywood J, Rannacher R, Turek S (1996) Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations. Int J Numer Methods Fluids 22(5):325–352
17. Meidner D, Rannacher R, Vihharev J (2009) Goal-oriented error control of the iterative solution of finite element equations. J Numer Math 17(2):143–172

18. Rannacher R (2000) Finite element methods for the incompressible Navier-Stokes equations. In: Galdi GP, Heywood JG, Rannacher R (eds) Fundamental directions in mathematical fluid mechanics. Birkhäuser, Basel, pp 191–293
19. Rannacher R (2004) Incompressible viscous flow. In: Stein E, de Borst R, Hughes TJR (eds) Encyclopedia of computational mechanics. Vol. 3. Fluids. Wiley, Chichester
20. Rannacher R, Vihharev J (2011) Adaptive finite element analysis of nonlinear problems: balancing of discretization and iteration error. Preprint, University of Heidelberg
21. Rannacher R, Westenberger A, Wollner W (2010) Adaptive finite element solution of eigenvalue problems: balancing of discretization and iteration error. J Numer Math 18(4):303–327
22. Saad Y (1980) Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices. Linear Algebra Appl 34:269–295
23. Schäfer M, Turek S (1996) Benchmark computations of laminar flow around a cylinder. In: Hirschel EH (ed) Flow simulation with high-performance computers II. NNFM, vol 52. Vieweg, Braunschweig, pp 547–566
24. Sorensen DC (2002) Numerical methods for large eigenvalue problems. Acta Numer 11:519–584
25. Verfürth R (1996) A review of a posteriori error estimation and adaptive Mesh-refinement techniques. Wiley-Teubner, Chichester

Chapter 7
On Quantitative Analysis of an Ill-Posed Elliptic Problem with Cauchy Boundary Conditions

Sergey Repin and Tuomo Rossi

Abstract In this paper, we consider an ill-posed boundary value problem for the equation $\operatorname{div} A\nabla u + f = 0$, which is closely connected with a problem of reconstruction of an unknown boundary condition. This problem can be reformulated as an unconstrained minimization problem for a convex nonnegative functional depending on the pair of variables (v, q), which approximate the desired solution and its flux, respectively. The functional vanishes if and only if v and q coincide with the exact solution of the problem (if the latter solution exists) and its flux, respectively. Moreover, we prove that if the functional is lesser than a small positive number ε, then ε-neighborhood of (v, q) contains the exact solution of the direct boundary value problem with mixed boundary conditions, which are traces on the boundary ε-close to the Cauchy conditions imposed. Advanced forms of the functional convenient for numerical computations are discussed.

Keywords Cauchy boundary conditions · Ill-posed problems · Inverse boundary value problems · Guaranteed error bounds

7.1 Introduction

Problems with overdetermined boundary conditions arise in technical applications related to identification of model parameters, scattering, reconstruction of images (see, e.g., [1–3, 6, 9] and the references therein), and also in fundamental problems in natural sciences (e.g., in astronomy). They are closely related to the theory of inverse problems, which mathematical foundations are well developed and presented in numerous publications (e.g., in the monographs [5, 13]). An overview of numerical methods can be found in, e.g., [4, 6]. The goal of this paper is to show (with the paradigm of a relatively simple elliptic problem) that new type a posteriori

S. Repin (✉) · T. Rossi
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland
e-mail: sergey.repin@jyu.fi

T. Rossi
e-mail: tuomo.rossi@jyu.fi

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_7,

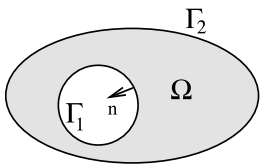


Fig. 7.1 Domain Ω with internal and external boundaries Γ_1 and Γ_2

error estimates derived in the last decade for direct boundary value problems can also be applied to inverse and ill-posed problems. More precisely, we select a model problem in 2-connected domain and show that the problem with overdetermined boundary conditions on one part of the boundary can be reformulated as a certain unconstrained minimization problem for a quadratic type functional. The latter functional attains minimal value (which is equal to zero provided that the problem data are compatible) only on the exact solution. Moreover, if the value of the functional is small for a pair of functions presenting an approximate solution and its flux, then we guarantee that a small neighborhood of this pair contains the exact solution of a direct problem with close boundary conditions.

We consider the elliptic operator

$$\mathscr{L}v = -\operatorname{div} A\nabla v,$$

where $A = \{a_{ij}\}$ is a symmetric matrix such that

$$a_{ij} \in L^{\infty}(\Omega), \quad i, j = 1, 2, \ldots, d, \tag{7.1}$$

$$c_1|\xi|^2 \le A\xi \cdot \xi \le c_2|\xi|^2, \quad c_2 \ge c_1 > 0. \tag{7.2}$$

Here, Ω is an open bounded set in $\mathbb{R}^d$ with boundary Γ consisting of two disjoint parts Γ_1 and Γ_2 (see Fig. 7.1). We are concerned with the following problem associated with the operator $\mathscr{L}$ (see, e.g., [7]): find $u \in H^1(\Omega)$ and $p \in H(\Omega, \operatorname{div})$ such that

$$-\operatorname{div} u = f \in L^2(\Omega), \quad \text{in } \Omega, \tag{7.3}$$

$$p = A\nabla u, \quad \text{in } \Omega, \tag{7.4}$$

$$u = u_0, \quad \text{on } \Gamma_1, \tag{7.5}$$

$$p \cdot n = g_0, \quad \text{on } \Gamma_1. \tag{7.6}$$

We assume that

$$\operatorname{meas}_{d-1} \Gamma_i > 0, \quad i = 1, 2 \tag{7.7}$$

and the Cauchy boundary conditions on Γ_1 are defined by the functions $u_0 \in H^{1/2}(\Gamma_1)$ and $g_0 \in L^2(\Omega)$.

This mathematical statement may arise in the theory of inverse problems if the data on Γ_1 are fully observable but the data on Γ_2 are unknown. In these problems, the goal is to reconstruct these unknown data using known boundary conditions

(7.5)–(7.6) and the fact that inside Ω the process is governed by the diffusion equation. More precisely, we wish to reconstruct u and $p \cdot n$ on Γ_2, what amounts solving the corresponding inverse problem.

It is known that problems like (7.3)–(7.6) are, in general, ill-posed (see, e.g., [5, 7]) and may have no solution if u_0 and g_0 are not coordinated. This fact may imply serious difficulties in numerical analysis of inverse problems. One way to minimize these difficulties is to reformulate (7.3)–(7.6) as an optimal control problem with differential constrains.

Problem $\mathscr{P}^\dagger$ Find $(u^\dagger, p^\dagger) \in W$ such that

$$J\big(u^\dagger, p^\dagger\big) = \inf J(v, q) := \int_{\Gamma_1} \big(|v - u_0|^2 + |q - g_0|^2\big) dx \to \min,$$

where infimum is seeking on the functions

$$(v, q) \in W := H^1(\Omega) \times H(\Omega), \quad H(\Omega) := \big\{q \in H(\Omega, \mathrm{div}),\ q \cdot n \in L^2(\Gamma_1)\big\}$$

satisfying the relations

$$\mathrm{div}\, q + f = 0, \quad \text{in } \Omega, \tag{7.8}$$

$$q = A \nabla v, \quad \text{in } \Omega. \tag{7.9}$$

Numerical methods based on Problem $\mathscr{P}^\dagger$ generate sequences $u_h^\dagger$ and $p_h^\dagger$, which can be used as approximations of u and p. Indeed, if $J(u^\dagger, p^\dagger) = 0$, then the required solution is found. However, in practical computations such a situation is unlikely and instead we obtain a pair of functions providing a small value of the cost functional J and satisfying (7.8)–(7.9) only approximately.

In this paper, we suggest other variational statements that can be used in numerical analysis of inverse problems. They are also defined on pairs of functions (v, q) and, in fact, mimic convex nonnegative functionals arising in the theory of functional a posteriori estimates (see [10, 12]). Using this theory, we prove two important properties of new variational statements. First, we show that infimum of the variational functional equals zero and it is attained only on the exact solution of (7.3)–(7.6). If this problem has no solution, then the functional remains positive for all admissible functions. Another useful property of the functional is that if the functional is smaller than ε, then small neighborhood of the respective pair (v, q) (which size is controllable and proportional to ε) contains exact solution of a certain direct boundary value problem with mixed boundary conditions that approximates the desired solution.

7.2 Error Measure for a Class of Boundary Value Problems

Together with the inverse problem $\mathscr{P}^\dagger$, we consider two direct boundary value problems $\mathscr{P}_1$ and $\mathscr{P}_2$.

Problem $\mathscr{P}_1$ Find $u \in H^1(\Omega)$ and $p \in H(\Omega, \mathrm{div})$ such that

$$\mathscr{L} u_1 = f \in L^2(\Omega), \quad \text{in } \Omega, \tag{7.10}$$

$$p_1 = A \nabla u_1, \quad \text{in } \Omega, \tag{7.11}$$

$$u_1 = \widetilde{u}, \quad \text{on } \Gamma_1, \tag{7.12}$$

$$p_1 \cdot n = \widehat{g}, \quad \text{on } \Gamma_2, \tag{7.13}$$

where $\widetilde{u} \in H^1(\Omega)$ and $\widehat{g} \in H(\Omega, \mathrm{div})$. The boundary conditions are understood in the generalized sense.

Problem $\mathscr{P}_2$ Find $u \in H^1(\Omega)$ and $p \in H(\Omega, \mathrm{div})$ such that

$$\mathscr{L} u_2 = f \in L^2(\Omega), \quad \text{in } \Omega, \tag{7.14}$$

$$p_2 = A \nabla u_2, \quad \text{in } \Omega, \tag{7.15}$$

$$u_2 = \widehat{u}, \quad \text{on } \Gamma_2, \tag{7.16}$$

$$p_2 \cdot n = \widetilde{g}, \quad \text{on } \Gamma_1, \tag{7.17}$$

where $\widehat{u} \in H^1(\Omega)$ and $\widetilde{g} \in H(\Omega, \mathrm{div})$.

Generalized solutions of these problems exists and belong to the sets $V_0^1 + \widetilde{u}$ and $V_0^2 + \widehat{u}$ (respectively), where

$$V_0^1 := \left\{ v \in H^1(\Omega) \mid v = 0 \text{ on } \Gamma_1 \right\}$$

and

$$V_0^2 := \left\{ v \in H^1(\Omega) \mid v = 0 \text{ on } \Gamma_2 \right\}.$$

They satisfy the integral identities

$$\int_\Omega A \nabla u_1 \cdot \nabla w \, dx = \int_\Omega f w \, dx + \int_{\Gamma_2} \widehat{g} w \, ds, \quad \forall w \in V_0^1, \tag{7.18}$$

$$\int_\Omega A \nabla u_2 \cdot \nabla w \, dx = \int_\Omega f w \, dx + \int_{\Gamma_2} \widetilde{g} w \, ds, \quad \forall w \in V_0^2. \tag{7.19}$$

Below we will derive computable estimates of the difference between u_1 (u_2) and any arbitrary function from the set $V_0^1 + \widetilde{u}$ ($V_0^2 + \widehat{u}$). For this purpose we apply the method developed in [12] with some changes necessary to adapt the corresponding estimates to the analysis of our inverse problem.

7.2.1 Estimates for Problem $\mathscr{P}_1$

Let

$$(v, q) \in \left(V_0^1 + \widetilde{u}\right) \times H(\Omega, \mathrm{div})$$

be an approximation of (u_1, p_1). In view of (7.18), we have

$$\int_\Omega A\nabla(u_1 - v) \cdot \nabla w \, dx = \int_\Omega (f w - A\nabla v \cdot \nabla w) \, dx + \int_{\Gamma_2} \widehat{g} w \, ds. \tag{7.20}$$

Let $q \in H(\Omega, \mathrm{div})$ by such that $q \cdot n \in L^2(\Gamma_2)$. Then,

$$\begin{aligned} &\int_\Omega A\nabla(u_1 - v) \cdot \nabla w \, dx \\ &\quad = \int_\Omega (f + \mathrm{div}\, q) w \, dx + \int_\Omega (q - A\nabla v) \cdot \nabla w \, dx + \int_{\Gamma_2} (q \cdot n - \widehat{g}) w \, ds. \end{aligned} \tag{7.21}$$

Let

$$\|\phi\|_{-1/2,\Gamma_2} := \sup_{w \in V_0^1} \frac{\int_{\Gamma_2} \phi w \, ds}{\|\nabla w\|},$$

where $\|\cdot\|$ denotes the L^2 norm in Ω.

Remark 7.1 If $\phi \in L^2(\Gamma_2)$, then $\|\phi\|_{-1/2,\Gamma_2} \le C_{tr}(\Omega, \Gamma_2)\|\phi\|_{\Gamma_2}$, where $C_{tr}(\Omega, \Gamma_2)$ is the trace embedding constant for the functions in V_0^1.

We set $w = u_1 - v$ in (7.21) and obtain

$$\begin{aligned} \left|\!\left|\!\left| \nabla(u_1 - v) \right|\!\right|\!\right| &\le |\!|\!| q - A\nabla v |\!|\!|_* + C_{\Omega\Gamma_1} \| \mathrm{div}\, q + f \| + \|q \cdot n - \widehat{g}\|_{-1/2,\Gamma_2} \\ &:= \mathfrak{M}_1(v, q), \end{aligned} \tag{7.22}$$

where

$$C_{\Omega\Gamma_1} := \sup_{w \in V_0^1} \frac{\int_\Omega |w|^2 dx}{\int_\Omega |\nabla w|^2 dx}.$$

In accordance with Remark 7.1, we estimate the last term of the majorant by a boundary integral provided that $q \cdot n \in L^2(\Gamma_2)$. Then, we obtain

$$\begin{aligned} \left|\!\left|\!\left| \nabla(u - v) \right|\!\right|\!\right| &\le |\!|\!| q - A\nabla v |\!|\!|_* + C_{\Omega\Gamma_1} \| \mathrm{div}\, q + f \| + C_{tr}(\Omega, \Gamma_2) \|q \cdot n - \widehat{g}\|_{\Gamma_2} \\ &:= \overline{\mathfrak{M}_1}(v, q). \end{aligned}$$

It is natural to measure the difference between (v, q) and (u_1, p_1) in terms of combined (primal-dual) norms

$$\left\|(v, q)\right\|_W := \|\nabla v\| + \|q\|_{\mathrm{div}} = \|\nabla v\| + \|q\| + \|\mathrm{div}\, q\|$$

and

$$\big\|(v,q)\big\|_W^{(1)} := |||\nabla v||| + |||q|||_* + C_{\Omega\Gamma_1}\|\operatorname{div} q\|,$$

which are obviously equivalent and define natural topology in W.

We have

$$\begin{aligned}\big\|(u_1-v,p_1-q)\big\|_W^{(1)} &= \big|||\nabla(u_1-v)|||\big| + |||p_1-q|||_* + C_{\Omega\Gamma_1}\|\operatorname{div} q + f\| \\ &\le 2\big|||\nabla(u_1-v)|||\big| + |||q - A\nabla v|||_* + C_{\Omega\Gamma_1}\|\operatorname{div} q + f\| \\ &\le 3\mathfrak{M}_1(v,q). \end{aligned} \tag{7.23}$$

It is easy to see that $\mathfrak{M}_1(v,q) = 0$ if and only if $v = u_1$ and $q = p_1$.

7.2.2 Estimates for Problem $\mathscr{P}_2$

Let

$$(v,q) \in \big(V_0^2 + \widehat{u}\big) \times H(\Omega, \operatorname{div})$$

be an approximation of (u_2, p_2). In view of (7.18), we have

$$\int_\Omega A\nabla(u_2 - v)\cdot\nabla w\,dx = \int_\Omega (fw - A\nabla v\cdot\nabla w)\,dx + \int_{\Gamma_1} \widetilde{g} w\,ds. \tag{7.24}$$

Let $q \in H(\Omega, \operatorname{div})$ by such that $q\cdot n \in L^2(\Gamma_1)$. Then

$$\begin{aligned}&\int_\Omega A\nabla(u_2 - v)\cdot\nabla w\,dx \\ &\quad= \int_\Omega (f + \operatorname{div} q) w\,dx + \int_\Omega (q - A\nabla v)\cdot\nabla w\,dx + \int_{\Gamma_1} (q\cdot n - \widetilde{g}) w\,ds. \end{aligned} \tag{7.25}$$

We set $w = u_1 - v$ in (7.21) and obtain

$$\big|||\nabla(u_2 - v)|||\big| \le |||q - A\nabla v|||_* + C_{\Omega\Gamma_2}\|\operatorname{div} q + f\| + \|q\cdot n - \widetilde{g}\|_{-1/2,\Gamma_1} := \mathfrak{M}_2(v,q), \tag{7.26}$$

where

$$C_{\Omega\Gamma_2} := \sup_{w\in V_0^2} \frac{\int_\Omega |w|^2 dx}{\int_\Omega |\nabla w|^2 dx}.$$

Now our goal is to estimate the error in terms of the combined norm

$$\big\|(v,q)\big\|_W^{(2)} := \|\nabla v\| + \|q\| + C_{\Omega\Gamma_2}\|\operatorname{div} q\|.$$

Analogously to the previous case, we have

$$\begin{aligned}\big\|(u_2-v,p_2-q)\big\|_W^{(2)} &= \big|\!\big|\!\big|\nabla(u_2-v)\big|\!\big|\!\big| + |\!|\!|p_2-q|\!|\!|_* + C_{\Omega\Gamma_2}\|\operatorname{div} q+f\| \\ &\le 2\big|\!\big|\!\big|\nabla(u_2-v)\big|\!\big|\!\big| + |\!|\!|q-A\nabla v|\!|\!|_* + C_{\Omega\Gamma_2}\|\operatorname{div} q+f\| \\ &\le 3\mathfrak{M}_2(v,q). \end{aligned} \tag{7.27}$$

It is easy to see that $\mathfrak{M}_2(v,q)=0$ if and only if $v=u_2$ and $q=p_2$.

If $q\cdot n\in L^2(\Gamma_1)$, then we obtain

$$\begin{aligned}\big|\!\big|\!\big|\nabla(u_2-v)\big|\!\big|\!\big| &\le |\!|\!|q-A\nabla v|\!|\!|_* + C_{\Omega\Gamma_2}\|\operatorname{div} q+f\| + C_{\mathrm{tr}}(\Omega,\Gamma_1)\|q\cdot n-\widetilde{g}\|_{\Gamma_1} \\ &:= \overline{\mathfrak{M}_2}(v,q).\end{aligned}$$

Remark 7.2 We note that

$$\gamma_1\big\|(v,q)\big\|_W \le \big\|(v,q)\big\|_W^{(1,2)} \le \gamma_2\big\|(v,q)\big\|_W, \tag{7.28}$$

where γ_1 and γ_2 depend on C_{Ω,Γ_1} and C_{Ω,Γ_2}. Therefore, (7.20) and (7.27) imply error bounds in terms of $\|\cdot\|_W$.

7.3 New Statements of the Basic Problem

Now we use the majorants derived in Sect. 2 in order to reformulate (7.3)–(7.6) as an unconstrained minimization problem. We prove that is the basic problem is solvable, then the new problem has the same solution.

7.3.1 Variational Statement 1

Let $I_1: W\to\mathbb{R}$ be defined by the relation

$$I_1(v,q) := \|A\nabla v-q\|_* + C_{\Omega\Gamma_1}\|\operatorname{div} q+f\|. \tag{7.29}$$

We consider the following variational problem generated by this convex and non-negative functional.

Problem $\mathscr{P}^{\ddagger_1}$ Find $(u^\ddagger,p^\ddagger)\in W_{\Gamma_1}$ such that

$$I_1\big(u^\ddagger,p^\ddagger\big) = \inf_{(v,q)\in W_{\Gamma_1}} I(v,q), \tag{7.30}$$

where

$$W_{\Gamma_1} := \big\{(v,q)\in W \mid v=u_0,\ q\cdot n=g_0 \text{ on } \Gamma_1\big\}.$$

Our goal is to show that this problem can be viewed as a generalized formulation of (7.3)–(7.6). Unlike $\mathscr{P}^\dagger$ it does not involve any differential constrains and, therefore, can be solved by direct minimization methods.

Theorem 7.1

(i) *The problem* (7.30) *has a solution.*
(ii) *If* $I_1(u^\ddagger, p^\ddagger) = 0$, *then this pair of functions solves* (7.3)–(7.6).
(iii) *Let* $(v_\varepsilon, q_\varepsilon) \in W$ *be such that*

$$I_1(v_\varepsilon, q_\varepsilon) \le \varepsilon. \tag{7.31}$$

Then, Problem $\mathscr{P}_1$ *with* $\widetilde{u} = u_0$ *and* $\widehat{g} = q_\varepsilon \cdot n$ *has the solution* $(u_1, p_1) \in W$ *such that*

$$\big\|(u_1 - v_\varepsilon, p_1 - q_\varepsilon)\big\|_W^{(1)} \le 3\varepsilon := \delta. \tag{7.32}$$

Proof Existence of the pair $(u^\ddagger, p^\ddagger)$ is guaranteed by general results of convex analysis (indeed W is a reflexive space and I is a nonnegative, convex, and coercive functional).

If $I(u^\ddagger, p^\ddagger) = 0$, then all the relations (7.3)–(7.6) are satisfied, so that $u^\ddagger = u$ and $p^\ddagger = p$. On the other hand, if there exists (u, p) satisfying (7.3)–(7.6), then $I(u, p) = 0$ and, consequently, these functions minimize the functional I (we recall that (u, p) belong to the set W_{Γ_1}).

We can view v_ε and q_ε as approximate solutions of Problem $\mathscr{P}_1$ with $\widetilde{u} = u_0$ and $\widehat{g} = q_\varepsilon \cdot n$. Then

$$\begin{aligned}\mathfrak{M}_1(v_\varepsilon, q_\varepsilon) &= \|A\nabla v_\varepsilon - q_\varepsilon\|_* + C_{\Omega\Gamma_1}\|\operatorname{div} q_\varepsilon + f\| + \|q_\varepsilon \cdot n - \widehat{g}\|_{-1/2,\Gamma_2} \\ &= \|A\nabla v_\varepsilon - q_\varepsilon\|_* + C_{\Omega\Gamma_1}\|\operatorname{div} q_\varepsilon + f\| \le \varepsilon.\end{aligned}$$

By (7.27), we obtain (7.32). □

Theorem 7.1 states that ε-neighborhood of $(v_\varepsilon, q_\varepsilon)$ contains exact solution of Problem $\mathscr{P}_1$ (which is a well posed boundary value problem with mixed Dirichlet-Neumann boundary conditions). It has the same Dirichlet boundary condition as the original problem (7.3)–(7.6). From (7.32) it follows that

$$\begin{aligned}\|p_1 \cdot n - g_0\|_{-1/2,\Gamma_1} &= \big\|(p_1 - q_\varepsilon) \cdot n\big\|_{-1/2,\Gamma_1} \\ &= \sup_{w \in V_0^2} \frac{\int_\Omega ((p_1 - q_\varepsilon) \cdot \nabla w + \operatorname{div}(p_1 - q_\varepsilon) w)dx}{\|\nabla w\|} \\ &\le \sqrt{1 + C_{\Omega\Gamma_1}^2}\,\|p_1 - q_\varepsilon\|_{\operatorname{div}} \le \mu = \sqrt{1 + C_{\Omega\Gamma_1}^2}\,\varepsilon.\end{aligned} \tag{7.33}$$

Thus, we can consider Problem $\mathscr{P}_1$ as a μ-approximation of the original problem (7.3)–(7.6) and use the boundary flux $p_1 \cdot n$ as an approximation of $p \cdot n$ on Γ_2.

Certainly, the trace $p_1 \cdot n$ on Γ_2 is unknown, but we can efficiently approximate it by $q_\varepsilon \cdot n$. Indeed,

$$\begin{aligned}&\big\|(p_1 - q_\varepsilon)\cdot n\big\|_{-1/2,\Gamma_2}\\ &\quad= \sup_{w\in V_0^1} \frac{\int_\Omega ((p_1 - q_\varepsilon)\cdot\nabla w + \operatorname{div}(p_1 - q_\varepsilon)w)dx}{\|\nabla w\|} \le \sqrt{1 + C^2_{\Omega\Gamma_2}}\,\varepsilon. \end{aligned} \tag{7.34}$$

Thus, the trace $q_\varepsilon \cdot n$ represents a good approximation of the normal flux $p_1 \cdot n$ on Γ_2.

In other words, if (7.31) holds, then on the external boundary Γ_2, the computable function $q_\varepsilon \cdot n$ represents $p_1 \cdot n$ associated with the direct boundary value problem, which satisfies (7.5) exactly and (7.6) with the accuracy μ. We note that in many cases sharp values of the boundary conditions on Γ_1 are not known because in real life problems all measurements are performed with some accuracy. Therefore, finding a pair (u_1, p_1) satisfying the boundary conditions with a certain accuracy may give a practically relevant answer.

Remark 7.3 Assume that the problem (7.3)–(7.6) has the solution (u, p). We have

$$\begin{aligned}\mu \ge \|p_1\cdot n - g_0\|_{-1/2,\Gamma_1} &= \big\|(p_1 - p)\cdot n\big\|_{-1/2,\Gamma_1}\\ &= \sup_{w\in V_0^2} \frac{\int_\Omega ((p_1 - p)\cdot\nabla w + (\operatorname{div} p_1 + f)w)dx}{\|\nabla w\|}. \end{aligned} \tag{7.35}$$

From (7.35) we find the estimate

$$\sup_{\substack{w\in V_0^2,\\ \|\nabla w\|=1}} \int_\Omega A\nabla(u_1 - u)\cdot\nabla w\,dx \le \mu, \tag{7.36}$$

which shows that u_1 approximates u at least in a weak sense.

7.3.2 Variational Statement 2

Let $I_2 : W \to \mathbb{R}$ be defined by the relation

$$I_2(v, q) := \|A\nabla v - q\|_* + C_{\Omega\Gamma_2}\|\operatorname{div} q + f\| + C_{tr}(\Omega, \Gamma_1)\|q\cdot n - g_0\|_{\Gamma_1}. \tag{7.37}$$

We consider the following variational problem generated by this convex and nonnegative functional.

Problem $\mathscr{P}^{\ddagger_2}$ Find $(u^\ddagger, p^\ddagger) \in W_{\Gamma_1}$ such that

$$I_2\big(u^\ddagger, p^\ddagger\big) = \inf_{\substack{v\in V_0^1 + u_0\\ q\in H(\Omega)}} I_2(v, q). \tag{7.38}$$

Theorem 7.2

(i) *The problem* (7.38) *has a solution.*
(ii) *If* $I_2(u^\ddagger, p^\ddagger) = 0$, *then the pair* $(u^\ddagger, p^\ddagger)$ *solves* (7.3)–(7.6).
(iii) *Let* $(v_\varepsilon, q_\varepsilon) \in W$ *be such that*

$$I_2(v_\varepsilon, q_\varepsilon) \le \varepsilon. \tag{7.39}$$

Then, Problem $\mathscr{P}_2$ *with* $\widehat{u} = v_\varepsilon$ *on* Γ_2 *and* $\widetilde{g} = q_\varepsilon \cdot n$ *on* Γ_1 *has the solution* $(u_2, p_2) \in W$ *such that*

$$\big\|(u_2 - v_\varepsilon, p_2 - q_\varepsilon)\big\|_W^{(2)} \le \delta. \tag{7.40}$$

Proof Existence of a minimizer follows from the same arguments that we have used in Theorem 7.1.

If $v = v_0$ on Γ_1 and, in addition, there exists $q \in H(\Omega, \mathrm{div})$ such that

$$A\nabla v = q \quad \text{and} \quad \mathrm{div}\, q + f = 0 \quad \text{in } \Omega$$

and

$$q \cdot n = g_0 \quad \text{on } \Gamma_1,$$

then (v, q) is the solution.

Let us view v_ε and q_ε as approximate solutions of Problem $\mathscr{P}_2$ with $\widehat{u} = v_\varepsilon$ and $\widetilde{g} = g_0$. We see that

$$\begin{aligned}\overline{\mathfrak{M}_2}(v_\varepsilon, q_\varepsilon) &:= \|A\nabla v_\varepsilon - q_\varepsilon\|_* + C_{\Omega\Gamma_2}\|\mathrm{div}\, q_\varepsilon + f\| + C_{tr}(\Omega, \Gamma_1)\|q_\varepsilon \cdot n - g_0\|_{\Gamma_1} \\ &\le \varepsilon \end{aligned} \tag{7.41}$$

and, therefore, (7.40) holds. □

Since $|||u_2 - v_\varepsilon||| \le \delta$, we conclude that

$$\|u_2 - u_0\|_{1/2,\Gamma_1} = \|u_2 - v_\varepsilon\|_{1/2,\Gamma_1} \le C_{tr}(\Omega, \Gamma_1)\|u_2 - v_\varepsilon\|_{H^1(\Omega)} \le \mathbb{C}_2\varepsilon, \tag{7.42}$$

where $\mathbb{C}_2 = 3\frac{\sqrt{1+C(\Omega\Gamma_2)}}{c_1} C_{tr}(\Omega, \Gamma_1)$. Hence, if (7.39) holds, then we guarantee that ε-neighborhood of $(v_\varepsilon, q_\varepsilon)$ contains exact solution of Problem $\mathscr{P}_2$, which satisfies (7.6) exactly and (7.5) approximately (with the accuracy $\mathbb{C}_2\varepsilon$).

7.4 Computational Aspects

For computations, it is more convenient to use squared forms of the functionals I_1 and I_2. For example, instead of I_1, we can minimize

$$\hat{I}_1(v, q; \beta) := (1 + \beta)\|A\nabla v - q\|_*^2 + C_{\Omega\Gamma_1}\frac{1+\beta}{\beta}\|\mathrm{div}\, q + f\|^2, \tag{7.43}$$

where β is an arbitrary positive number. It is easy to see that for any $\beta > 0$

$$\big(I_1(v,q)\big)^2 \le \hat{I}_1(v,q;\beta) \tag{7.44}$$

so that small values of $\hat{I}_1$ imply small values of I_1. Moreover, $\hat{I}_1$ vanishes only on the exact solution (as I_1). At the same time minimization of a quadratic functional is a simpler task. (In particular, it can be reduced to solving systems of linear simultaneous equations with the help of efficient methods of numerical linear algebra.)

However, using this procedure requires the constant $C_{\Omega\Gamma_1}$ (or a good upper bound of it). If Ω is a domain with complex boundaries, then finding $C_{\Omega\Gamma_1}$ may be a difficult problem. Below we show a way to bypass this difficulty and obtain a computable upper bound of I_1, which preserves the main properties of this functional and does not involve unknown constants. For this purpose, we use advanced forms of the functional error majorants (see [12]).

Assume that $\bar{\Omega} = \bigcup_{i=1}^{N} \bar{\Omega}_i$, where Ω_i are nonintersecting domains with Lipschitz continuous boundaries. We impose additional requirements on q, namely

$$\int_{\Omega_i} (\operatorname{div} q + f)dx = 0, \quad i = 1, 2, \dots, N. \tag{7.45}$$

In this case, instead of I_1, we have another functional that contains constants $C_{P\Omega_i}$, $i = 1, 2, \dots, N$ instead of $C_{\Omega\Gamma_1}$, which are the constants in the Poincaré inequality for Ω_i. This functional is generated by the estimate

$$\big|\!\big|\!\big|\nabla(u_1 - v)\big|\!\big|\!\big|^2 \le |||A\nabla v - q|||_* + \sqrt{\sum_{i=1}^{N} C_{P\Omega_i}^2 \,\|\operatorname{div} q + f\|_{\Omega_i}^2}. \tag{7.46}$$

If Ω_i is a convex domain, then (see [11])

$$C_{P\Omega_i} \le \frac{\operatorname{diam}\Omega_i}{\pi}.$$

In this case, the right hand side of (7.46) is fully computable and we can reformulate Problem $\mathscr{P}_1^{\ddagger}$ as follows: find $(u^{\ddagger}, p^{\ddagger}) \in W_{\Gamma_1}$ such that

$$\bar{I}_1\big(u^{\ddagger}, p^{\ddagger}\big) = \inf_{(v,q)\in W_{\Gamma_1}} I(v,q), \tag{7.47}$$

where

$$\bar{I}_1(v,q) := |||A\nabla v - q|||_* + \sqrt{\sum_{i=1}^{N} \left(\frac{\operatorname{diam}\Omega_i}{\pi}\right)^2 \|\operatorname{div} q + f\|_{\Omega_i}^2}.$$

It is not difficult to prove that this problem possesses the same properties as Problem $\mathscr{P}_1^{\ddagger}$, so that if $(v_\varepsilon, q_\varepsilon)$ is a pair such that $\bar{I}_1(v,q) \le \varepsilon$, then exact solution

(u_1, p_1) of a direct boundary value problem with mixed boundary conditions lies in ε-neighborhood of $(v_\varepsilon, q_\varepsilon)$.

For practical minimization, it is convenient to operate with the squared functional, which in this case has the form

$$\hat{\hat{I}}_1(v, q; \beta) = (1+\beta)|||A\nabla v - q|||_*^2 + \frac{1+\beta}{\beta}\sum_{i=1}^{N} C_{P\Omega_i}^2 \| \operatorname{div} q + f\|_{\Omega_i}^2. \quad (7.48)$$

Finally, we note that physically motivated statements of inverse problems associated with diffusion models typically operate with not fully defined data (e.g., coefficients of A and/or source term f). The functionals I_1, I_2, and $\hat{\hat{I}}_1$ have the structure that allows us to account data indeterminacy in the process of finding approximate solutions of inverse problems by means of the same techniques that has been developed in [8] for direct boundary value problems.

References

1. Bristeau MO, Glowinski R, Périaux J (1993) Numerical simulation of high frequency scattering waves using exact controllability methods. In: Nonlinear hyperbolic problems: theoretical, applied, and computational aspects, Taormina, 1992. Notes numer fluid mech, vol 43. Vieweg, Braunschweig, pp 86–108
2. Chen HQ, Glowinski R, Périaux J (2008) A domain decomposition/Nash equilibrium methodology for the solution of direct and inverse problems in fluid dynamics with evolutionary algorithms. In: Domain decomposition methods in science and engineering XVII. Lect notes comput sci eng, vol 60. Springer, Berlin, pp 21–32
3. Egger H, Schlottbom M (2011) Efficient reliable image reconstruction schemes for diffuse optical tomography. Inverse Probl Sci Eng 19(2):155–180
4. Engl HW (2000) Inverse problems and their regularization. In: Computational mathematics driven by industrial problems, Martina Franca, 1999. Lecture notes in math, vol 1739. Springer, Berlin, pp 127–150
5. Isakov V (2006) Inverse problems for partial differential equations, 2nd edn. Applied mathematical sciences, vol 127. Springer, New York
6. Kügler P, Engl HW (2002) Identification of a temperature dependent heat conductivity by Tikhonov regularization. J Inverse Ill-Posed Probl 10(1):67–90
7. Lattès R, Lions J-L (1967) Méthode de quasi-réversibilité et applications. Dunod, Paris
8. Mali O, Repin S (2010) Two-sided estimates of the solution set for the reaction-diffusion problem with uncertain data. In: Applied and numerical partial differential equations. Comput methods appl sci, vol 15. Springer, New York, pp 183–198
9. Natterer F (2008) X-ray tomography. In: Inverse problems and imaging. Lecture notes in math, vol 1943. Springer, Berlin, pp 17–34
10. Neittaanmäki P, Repin S (2004) Reliable methods for computer simulation. Error control and a posteriori estimates. Elsevier, Amsterdam
11. Payne LE, Weinberger HF (1960) An optimal Poincaré inequality for convex domains. Arch Ration Mech Anal 5:286–292
12. Repin S (2008) A posteriori estimates for partial differential equations. Walter de Gruyter, Berlin
13. Tarantola A (1987) Inverse problem theory. Elsevier, Amsterdam

Chapter 8
On the Advantages and Drawbacks of A Posteriori Error Estimation for Fourth-Order Elliptic Problems

Karel Segeth

Abstract In this survey contribution, we present and compare, from the viewpoint of adaptive computation, several recently published error estimation procedures for the numerical solution of biharmonic and some further fourth order elliptic problems mostly in 2D. In the *hp*-adaptive finite element method, there are two possibilities to assess the error of the computed solution a posteriori: to construct a classical analytical error estimate or to obtain a more accurate reference solution by the same procedure as the approximate solution and, from it, the computational error estimate. For the lack of space, we sometimes only refer to the notation introduced in the papers quoted. The complete hypotheses and statements of the theorems presented should also be looked for there.

8.1 Introduction

Numerical computation has always been connected with some control procedures. It means that the approximate result is of primary importance, but also the error of this computed result, i.e. some norm of the difference between the exact and approximate solution brings important information. The exact solution is usually not known. This means that we can get only some estimates of the error.

The development of numerical procedures has been accompanied with *a priori error estimates* that are very useful in theory but usually include constants that are completely unknown, in better cases can be estimated. In particular, the development of the finite element method, and its *h*-version and *hp*-version required reliable and computable estimates of the error that depend only on the approximate solution just computed, if possible. This is the means for the local mesh refinement in the *h*-version and, moreover, also for the increase of the polynomial degree in the *p*-version.

K. Segeth (✉)
Institute of Mathematics, Academy of Sciences, Prague, Czech Republic
e-mail: segeth@math.cas.cz

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_8,

We employ a quantity called the *a posteriori error indicator* η_T for all triangles T of the triangulation $\mathscr{T}_h$ and, if not defined otherwise, the *error estimator*

$$\varepsilon = \sqrt{\sum_{T \in \mathscr{T}_h} \eta_T^2},$$

see [5], in each of the estimation strategies that follow to assess the error of the approximate solution. The quality of an a posteriori error estimator is often measured by its *effectivity index*, i.e. the ratio of some norm of the error estimate and the true error. An error estimator is called *effective* if both its effectivity index and the inverse of the index remain bounded for all meshsizes of triangulations. It is called *asymptotically exact* if its effectivity index converges to 1 as the meshsize tends to 0.

Undoubtedly, obtaining efficient and computable a posteriori error estimates is not easy. (Note that *computable* means, among others, that the degree of piecewise polynomials approximating the solution is high enough.) The papers [2, 3] by Babuška and Rheinboldt represent the pioneering work in this field. The books [1, 4] are surveys of the state of the art some time ago while [17] is an attempt to compare some a posteriori error estimators.

There are several classes of a posteriori error indicators and estimators based on different approaches and their names slightly vary in the literature. We consider residual or recovery a posteriori error indicators for the solution of the biharmonic equation in the classical weak formulation [19, 20] and in the Ciarlet-Raviart formulation [8, 12] in Sect. 8.3. We further present recovery or residual a posteriori error indicators for the solution of a more general 4th order equation [6, 14] and, in particular, functional error estimators [11, 13, 16] in Sect. 8.4. Section 8.5 is devoted to a brief conclusion.

8.2 Notation and Preliminaries

A common notation is introduced in this section. We write $C(S)$ for the space of all functions continuous on the set S, $C_m(S)$ for that of all functions continuous together with their m derivatives.

Let $\Omega \subset R^n$, $n \geq 1$, be a bounded domain (i.e. a bounded connected open set) with the boundary Γ. We use the obvious notation for the $L_2(\Omega)$, $L_\infty(\Omega)$, $H^1(\Omega)$ and $H^2(\Omega)$ norms, and for the $H^k(\Omega)$ seminorm. Let $\Phi = [\varphi_{ik}]$ and $\Psi = [\psi_{ik}]$ be $n \times n$ matrices, $\Phi, \Psi \in R^{n \times n}$. We introduce their *elementwise matrix product* $\Phi \odot \Psi \in R$ and the *Frobenius* or *Schur norm* of the matrix Φ as $\|\Phi\|_{\mathrm{F}} = \sqrt{\Phi \odot \Phi}$.

The norm or seminorm may be restricted to any open set $\omega \subset \Omega$ with the Lipschitz boundary γ. We thus write, e.g., $\|\cdot\|_{0;\omega}$ for the $L_2(\omega)$ norm. We also employ the spaces $H_0^1(\Omega)$, $H_0^2(\Omega)$, etc. and the adjoint spaces $H^{-k}(\Omega)$, $k > 0$, of linear functionals. We often omit the symbol Ω if Ω is the domain concerned.

Let V be a real Hilbert space and $a : V \times V \to R$ a bounded symmetric coercive bilinear form. The energy norm induced by this bilinear form is denoted by

$$|||v||| = \sqrt{a(v, v)}. \tag{8.1}$$

We use the notation

$$\operatorname{div} A = \nabla \cdot A = \sum_{s=1}^{n} \frac{\partial a_s}{\partial x_s} \in R$$

for the divergence of a differentiable vector-valued function $A = [a_1, \dots, a_n]$. We put $\nabla A = \nabla \otimes A \in R^{n\times n}$, where $\otimes$ is the tensor product, for the vector-valued function A and $\nabla b = \operatorname{grad} b \in R^n$ for the gradient of a differentiable scalar-valued function b. Furthermore, for a differentiable matrix-valued function $\Theta = [\vartheta_{ij}]_{i,j=1}^{n}$ we introduce its divergence as a vector-valued function

$$\operatorname{Div} \Theta = \nabla \cdot \Theta = \sum_{j=1}^{n} \frac{\partial \vartheta_{ij}}{\partial x_j} \in R^n.$$

Let $R_{\mathrm{s}}^{n\times n}$ be the space of real symmetric $n \times n$ matrices. We consider also the space $H(\operatorname{div}, \Omega) = \{Y \in L_2(\Omega, R^n) \mid \operatorname{div} Y \in L_2(\Omega)\}$ of vector-valued functions Y and the space $H(\operatorname{Div}, \Omega) = \{\Theta \in L_2(\Omega, R_{\mathrm{s}}^{n\times n}) \mid \operatorname{Div} \Theta \in L_2(\Omega, R^n)\}$ of symmetric matrix-valued functions Θ.

For a matrix-valued function $\Phi : \Omega \to R^{n\times n}$, $\Phi = [\varphi_{ik}]$, we put

$$\operatorname{div}^2 \Phi = \sum_{i=1}^{n} \sum_{k=1}^{n} \frac{\partial^2 \varphi_{ik}}{\partial x_i \partial x_k} \in R$$

provided these derivatives exist.

Finally, let

$$H\big(\operatorname{div}^2, \Omega\big) = \big\{\Phi \in L_2\big(\Omega, R^{n\times n}\big) \mid \operatorname{div}^2 \Phi \in L_2(\Omega)\big\},$$
$$H(\operatorname{div}\operatorname{Div}, \Omega) = \big\{\Phi \in L_2\big(\Omega, R_{\mathrm{s}}^{n\times n}\big) \mid \operatorname{div}\operatorname{Div} \Phi \in L_2(\Omega)\big\}$$

be the spaces of matrix-valued and symmetric matrix-valued functions, respectively.

Symbols $c, c_1, \dots$ are generic. They may represent different quantities (depending possibly on other different quantities) at different occurrences.

8.2.1 Finite Element Mesh Notation

Let $\mathscr{F} = \{\mathscr{T}_h \mid h > 0\}$ be a family of triangulations $\mathscr{T}_h$ of Ω. For any triangle $T \in \mathscr{T}_h$ we denote by h_T its diameter, while h indicates the maximum size of all the triangles in the mesh. We further denote by ϱ_T the diameter of the largest ball inscribed into T. Let $\mathscr{E}(T)$ be the set of all edges and $\mathscr{N}(T)$ the set of all nodes of T. We set

$$\mathscr{E}_h = \bigcup_{T\in\mathscr{T}_h} \mathscr{E}(T), \qquad \mathscr{N}_h = \bigcup_{T\in\mathscr{T}_h} \mathscr{N}(T).$$

We split $\mathscr{E}_h$ in the form $\mathscr{E}_h = \mathscr{E}_{h,\Omega} \cup \mathscr{E}_{h,\Gamma}$ with

$$\mathscr{E}_{h,\Omega} = \{E \in \mathscr{E}_h \mid E \subset \Omega\}, \qquad \mathscr{E}_{h,\Gamma} = \{E \in \mathscr{E}_h \mid E \subset \Gamma\}.$$

For $T \in \mathscr{T}_h$ we define

$$\omega_T = \bigcup_{\mathscr{E}(T)\cap\mathscr{E}(T')\neq\emptyset} T'.$$

The length of $E \in \mathscr{E}_h$ is denoted by h_E. Finally, with every edge $E \in \mathscr{E}_h$ we associate a unit normal vector n_E. The choice of the outer direction of n_E is arbitrary but fixed.

Let T_+ and T_- be any two triangles with a common edge $E \in \mathscr{E}_{h,\Omega}$, the subscripts $+$ and $-$ being chosen in such a way that the unit outer normal to T_- at E corresponds to n_E. Given a piecewise continuous scalar-valued function w on Ω, call w^+ or w^- its trace $w|_{T_+}$ or $w|_{T_-}$ on E. The jump of w across E in the direction of n_E is given by

$$[w]_E = w^+ - w^-.$$

The jump across an edge from $\mathscr{E}_{h,\Gamma}$ is simply given by the trace of the function w on the edge (i.e., the value of w outside Ω is assumed to be zero). For a vector-valued function, the jump is defined componentwise.

We further write $P_l(T)$ for the space of polynomials of degree at most l on T, $l \geq 0$ fixed. In the sequel, $\pi_{l,T}$ denotes the L_2 orthogonal projection of $L_1(T)$ onto $P_l(T)$.

Finally, let f_h be an approximation of a function $f \in L_2(\Omega)$ on a triangle $T \in \mathscr{T}_h$. We then put

$$e_T = \|f - f_h\|_{0;T}. \tag{8.2}$$

8.3 Dirichlet and Second Problems for Biharmonic Equation

8.3.1 Dirichlet Problem for Biharmonic Equation

Let the domain $\Omega \subset R^2$ have a polygonal boundary Γ. We consider the two dimensional biharmonic problem

$$\Delta^2 u = f \quad \text{in } \Omega, \tag{8.3}$$

$$u = \frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma \tag{8.4}$$

with $f \in L_2(\Omega)$ that models, e.g., the vertical displacement of the mid-surface of a clamped plate subject to bending.

Let X and Y be two Banach spaces with norms $\|\cdot\|_X$ and $\|\cdot\|_Y$. Let $\mathscr{L}(X, Y)$ denote the Banach space of continuous linear maps of X on Y and $\mathrm{Isom}(X, Y) \subset$

$\mathscr{L}(X, Y)$ an open subset of linear homeomorphisms of X onto Y. Let $Y^* = \mathscr{L}(Y, R)$ be the dual space of Y and $\langle \cdot, \cdot \rangle$ the corresponding duality pairing.

Let us put, in particular,

$$X = Y = H_0^2(\Omega), \qquad \|\cdot\|_X = \|\cdot\|_Y = \|\cdot\|_2,$$
$$\langle F(u), v\rangle = \int_\Omega \Delta u \Delta v - \int_\Omega f v. \tag{8.5}$$

We then say that $u \in X$ is the weak solution of the problem (8.3), (8.4) if

$$\langle F(u), v\rangle = 0 \tag{8.6}$$

for all $v \in Y$.

Since the bilinear form

$$\{u, v\} \to \int_\Omega \Delta u \Delta v$$

is continuous and coercive on X (cf. [9]), we have $\mathrm{d}F(u) \in \mathrm{Isom}(X, Y^*)$ for all $u \in X$, where $\mathrm{d}F$ is the derivative.

Let $\mathscr{F} = \{\mathscr{T}_h \mid h > 0\}$ be a regular family of triangulations $\mathscr{T}_h$ of Ω (see, e.g., [9]). For the discretization of the problem (8.3), (8.4) we assume that $X_h \subset X$ and $Y_h \subset Y$ are finite element spaces corresponding to $\mathscr{T}_h$ and consisting of piecewise polynomials. These conditions imply in particular that the functions in X_h and Y_h are of class C_1. Denote by k, $k \geq 1$, the maximum polynomial degree of the functions in X_h. Further, put $f_h = \pi_{l,T} f$ on T for a fixed $l \geq 0$.

Replacing f in the definition (8.5) by f_h to get the functional F_h, we say that $u_h \in X_h$ is the approximate solution of the problem (8.3), (8.4) if

$$\langle F_h(u_h), v_h\rangle = 0 \tag{8.7}$$

for all $v_h \in Y_h$.

Using the notation (8.2) for e_T and defining the *local residual a posteriori error indicator*

$$\eta_{\mathrm{V},T} = \Bigg(h_T^4 \left\| \Delta^2 u_h - f_h \right\|_{0;T}^2 + \sum_{E \in \mathscr{E}(T) \cap \mathscr{E}_{h,\Omega}} \left(h_E \left\| [\Delta u_h]_E \right\|_{0;E}^2 + h_E^3 \left\| [n_E \cdot \nabla \Delta u_h]_E \right\|_{0;E}^2 \right) \Bigg)^{1/2}$$

for all $T \in \mathscr{T}_h$, we have the following theorem [19].

Theorem 8.1 *Let $u \in X$ be the unique weak solution of the problem (8.3), (8.4), i.e. of (8.6), and let $u_h \in X_h$ be an approximate solution of the corresponding discrete*

problem (8.7). Then we have the a posteriori estimates

$$\|u-u_h\|_2 \le c_1\varepsilon_{\mathrm{V}} + c_2\left(\sum_{T\in\mathscr{T}_h} h_T^4 e_T^2\right)^{1/2} + c_3\left\|F(u_h)-F_h(u_h)\right\|_{Y_h^*} + c_4\left\|F_h(u_h)\right\|_{Y_h^*}$$

and

$$\eta_{\mathrm{V},T} \le c_5\|u-u_h\|_{2;\omega_T} + c_6\left(\sum_{T'\subset\omega_T} h_{T'}^4\varepsilon_{T'}^2\right)^{1/2}$$

for all $T\in\mathscr{T}_h$. The quantities $\|F(u_h)-F_h(u_h)\|_{Y_h^*}$ and $\|F_h(u_h)\|_{Y_h^*}$ represent the consistency error of the discretization and the residual of the discrete problem, and the quantities $c_1,\ldots,c_6$ may depend only on h_T/ϱ_T, and the integers k and l.

The proof is given in [19]. It seems that this is the first a posteriori error estimate for 4th order problems published.

Let us now consider a nonconforming approximate solution. We say that the family $\mathscr{F}=\{\mathscr{T}_h \mid h>0\}$ of triangulations $\mathscr{T}_h$ is *shape regular* if there are positive constants r_1 and r_2 such that for each triangle $T\in\mathscr{T}_h$ we may inscribe a ball of radius r_1h_T in T and inscribe T in a ball of radius r_2h_T. Thus, let $\mathscr{F}$ be a shape regular family of triangulations $\mathscr{T}_h$ of Ω. Letting T_x be an arbitrary triangle containing the point x, we denote by $h(x)$ the diameter of the triangle T_x.

Let (T, P_T, Φ_T) be the Zienkiewicz element with the triangle $T\in\mathscr{T}_h$, the shape function space P_T, and the set of nodal parameters Φ_T consisting of the function values and two values of first-order derivatives at the three vertices of T [9]. This element is sometimes called the TQC9 element and the corresponding finite element approximation of the fourth-order problem (8.3), (8.4) is nonconforming.

Corresponding to $\mathscr{T}_h$, denote by V_h and V_{h0} the above introduced Zienkiewicz element spaces with respect to H^2 and H_0^2, respectively. For $u_h\in V_h$ and $T\in\mathscr{T}_h$, we define the *local residual a posteriori error indicators* $\eta_{\mathrm{W},T}$ and $\widetilde{\eta}_{\mathrm{W},T}$ like in [20]. The corresponding statement proven there yields two a posteriori error estimates that contain unknown positive constants C_1 and C_2.

8.3.2 Dirichlet and Second Problems for Biharmonic Equation in Mixed Finite Element Formulation

Let $\Omega\subset R^2$ be a convex polygonal domain with the boundary Γ. We consider the two-dimensional biharmonic problem

$$\Delta^2 u = f \quad \text{in } \Omega, \tag{8.8}$$

$$u = \frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma \tag{8.9}$$

with $f \in H^{-1}(\Omega)$ that is used both for linear plate analysis and incompressible flow simulation.

Put $V = H_0^1(\Omega)$ and $X = H^1(\Omega)$ and define the continuous bilinear forms

$$a(w,z) = \int_\Omega wz \quad \text{on } X \times X \quad \text{and} \quad b(z,u) = \int_\Omega \nabla z \cdot \nabla u \quad \text{on } X \times V \tag{8.10}$$

with scalar-valued functions u, w, and z.

The Ciarlet-Raviart weak formulation [10] of (8.8) and (8.9) then reads: Find $\{w,u\} \in X \times V$ such that

$$a(w,z) + b(z,u) = 0 \quad \text{for all } z \in X, \tag{8.11}$$

$$b(w,v) + \int_\Omega fv = 0 \quad \text{for all } v \in V. \tag{8.12}$$

The existence and uniqueness of the solution $\{w = \Delta u, u\}$ of the problem (8.11) and (8.12) are proven in [7].

We construct the conforming second order discretization according to [15]. Let $\mathscr{F} = \{\mathscr{T}_h \mid h > 0\}$ be a regular family of triangulations $\mathscr{T}_h$ of Ω. For the sake of simplicity, we also assume that the family is uniformly regular [9] to guarantee that the inequality (8.13) holds, even though it is not easy to satisfy this condition in the presence of mesh refinements.

The finite element spaces X_h and V_h are then

$$X_h = \left\{x_h \in X \mid x_h|_T \in P_2(T) \text{ for all } T \in \mathscr{T}_h\right\},$$
$$V_h = \left\{v_h \in V \mid v_h|_T \in P_2(T) \text{ for all } T \in \mathscr{T}_h\right\}.$$

Our assumption of uniform regularity of the family $\mathscr{F}$ implies that there is a positive constant c such that the *inverse inequality*

$$|x_h|_{m;T} \le ch^{l-m}|x_h|_{l;T} \tag{8.13}$$

holds for all integers l and m, $l \le m$, and all $x_h \in X_h$ and $T \in \mathscr{T}_h$.

The discrete formulation of the problem (8.11) and (8.12) now reads: Find $\{w_h, u_h\} \in X_h \times V_h$ such that

$$a(w_h, z_h) + b(z_h, u_h) = 0 \quad \text{for all } z_h \in X_h, \tag{8.14}$$

$$b(w_h, v_h) + \int_\Omega fv_h = 0 \quad \text{for all } v_h \in V_h. \tag{8.15}$$

We introduce the *local residual a posteriori error indicators* $\eta_{C,T}$ and $\widetilde{\eta}_{C,T}$ based on local residuals like in [8]. Then the following theorem holds.

Theorem 8.2 *Let $\{w,u\} \in X \times V$ be the unique weak solution of the problem (8.8) and (8.9), i.e. of (8.11) and (8.12), and let $\{w_h, u_h\} \in X_h \times V_h$ be an approximate*

solution of the corresponding discrete problem (8.14) *and* (8.15). *Then we have the a posteriori estimates*

$$\|u - u_h\|_1 + h\|w - w_h\|_0 \le C_1\left(\varepsilon_C + h^2\widetilde{\varepsilon}_C\right)$$

with some positive constant C_1 *independent of* h *and*

$$\eta_{C,T} + h_T^2\widetilde{\eta}_{C,T} \le C_2\left(|u - u_h|_{1;\omega_T} + h_T\|w - w_h\|_{0;\omega_T} + h_T^3\sum_{T'\subset\omega_T} e_{T'}\right)$$

for $T \in \mathscr{T}_h$ *with some positive constant* C_2 *independent of* h *and* e_T *given by* (8.2).

The proof is given in [8].

On the convex polygonal domain $\Omega \subset R^2$ with the boundary Γ we now consider the two dimensional second biharmonic problem

$$\Delta^2 u = f \quad \text{in } \Omega, \tag{8.16}$$

$$u = \Delta u = 0 \quad \text{on } \Gamma \tag{8.17}$$

with $f \in L_2(\Omega)$ that models the deformation of a simply supported thin elastic plate. Putting $w = \Delta u$, we can rewrite the problem (8.16), (8.17) as the system of two Poisson equations, both with the homogeneous Dirichlet boundary condition.

Define the continuous bilinear forms $a(w, z)$ and $b(z, u)$ by (8.10) but with all the scalar-valued functions u, w, and z from $V = H_0^1(\Omega)$. The Ciarlet-Raviart weak formulation [10] of (8.16) and (8.17) then reads: Find $\{w, u\} \in V \times V$ such that (8.11) and (8.12) hold for all $z, v \in V$.

Let $\mathscr{F} = \{\mathscr{T}_h \mid h > 0\}$ be a quasiuniform family of triangular or rectangular partitions $\mathscr{T}_h$ of Ω [1]. Put

$$V_h = \left\{z \in C(\overline{\Omega}) \mid z|_T \in P_k(T),\ k \ge 1,\ \text{for all } T \in \mathscr{T}_h\right\} \cap H_0^1(\Omega).$$

The discrete weak formulation of the problem (8.16) and (8.17) now reads: Find $\{w_h, u_h\} \in V_h \times V_h$ such that (8.14) and (8.15) hold for all $z_h, v_h \in V_h$.

Let the basis function $v_{h,N}$ from V_h be associated with the node $N \in \mathscr{N}_{h,\Omega} = \mathscr{N}_h \cap \Omega$. Put $\omega_N = \operatorname{supp} v_{h,N}$. We introduce the *gradient recovery operator* $G v_h : V_h \to V_h \times V_h$ in the following way [12]. Assume that

$$v_h(x) = \sum_{N\in\mathscr{N}_{h,\Omega}} \beta_N v_{h,N}(x), \quad x \in \Omega,$$

with some coefficients β_N and put

$$\widetilde{G} v_{h,N} = \sum_{T\cap\omega_N\neq\emptyset} \alpha_N^T(\nabla v_{h,N})|_T, \quad \text{where} \sum_{T\cap\omega_N\neq\emptyset} \alpha_N^T = 1$$

and $0 \le \alpha_N^T \le 1$ are chosen weights. Note that the vector $\nabla v_{h,N}$ is constant on each triangle. Finally, we set

$$Gv_h(x) = \sum_{N \in \mathcal{N}_{h,\Omega}} \widetilde{G}v_{h,N}\ v_{h,N}(x), \quad x \in \Omega.$$

For $u_h, w_h \in V_h$ and $T \in \mathcal{T}_h$, define a *local recovery a posteriori error indicator* $\eta_{\mathrm{L},T}$ like in [12]. The corresponding statement proven there yields a lower as well as an upper a posteriori error estimate that both contain unknown positive constants c, C, C_1, and C_2 independent of h. In the paper, the authors further claim that the global error estimator ε_{L} is asymptotically exact if the mesh is uniform and the solution is smooth enough.

8.4 Dirichlet Problem for Fourth Order Elliptic Equation

8.4.1 Some Recovery and Residual Error Indicators

Put $\Omega = (0, 1) \subset R^1$. Let all the functions concerned be scalar-valued functions of a single variable. We consider the one dimensional boundary value problem for the ordinary fourth-order elliptic equation

$$\left(au''\right)'' = f \quad \text{in } \Omega$$

with the boundary conditions

$$u(0) = u'(0) = 0, \qquad u(1) = u'(1) = 0.$$

The weak solution $u \in H_0^2(\Omega)$ and the approximate solution $u_h \in V_h$ are defined in the usual way [14]. V_h is a finite element space consisting of piecewise Hermite cubic polynomials.

We introduce a *recovery operator* Gv_h for the second derivative of $v_h \in V_h$ and, for $u_h \in V_h$ and $T \in \mathcal{T}_h$, define a *local recovery a posteriori error indicator* $\eta_{\mathrm{P},T}$ like in [14]. The corresponding statement proven there yields an upper estimate for the difference of the global error estimator ε_{P} and the energy norm of the true error [14]. The global error estimator is asymptotically exact.

Consider the bending problem of an isotropic linearly elastic plate. The bilinear form for the problem is

$$a(u, v) = \left(\gamma\varepsilon(\nabla u), \varepsilon(\nabla v)\right)_0, \quad u, v \in H_0^2,$$

where γ is the fourth-order positive definite *elasticity tensor* and ε the *small strain tensor* [6]. We employ the *discrete Morley space* W_h that is nonconforming for the finite element solution of the problem, see, e.g., [9].

With the help of the bilinear form $a_h(u_h, v_h)$, $v_h \in W_h$, defined in an obvious way we introduce the approximate solution $u_h \in W_h$. The bilinear form a_h is positive definite on the space W_h, therefore there is a unique solution $u_h \in W_h$ to the problem, cf. [6].

For $u_h \in W_h$ and $T \in \mathscr{T}_h$, define a *local residual a posteriori error indicator* $\eta_{\mathrm{B},T}$ like in [6]. The corresponding statement proven there yields lower as well as upper a posteriori error estimates in a discrete norm introduced there. Both these estimates contain an unknown positive constant C independent of h.

8.4.2 Dirichlet Problem for Fourth Order Partial Differential Equation

Let $\Omega \in R^n$ be a bounded connected domain and Γ its Lipschitz continuous boundary. We consider the 4th order elliptic problem for a scalar-valued function u,

$$\operatorname{div}\operatorname{Div}(\gamma \nabla\nabla u) = f \quad \text{in } \Omega, \tag{8.18}$$

$$u = \frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma, \tag{8.19}$$

where $f \in L_2(\Omega)$, $\gamma = [\gamma_{ijkl}]_{i,j,k,l=1}^n$ and $\gamma_{ijkl} = \gamma_{jikl} = \gamma_{klij} \in L_\infty(\Omega)$.

We assume the existence of constants $0 < m \le M$ such that

$$m\|\Phi\|_{\mathrm{F}}^2 \le (\gamma\Phi) \odot \Phi \le M\|\Phi\|_{\mathrm{F}}^2 \quad \text{for all } \Phi \in R_{\mathrm{s}}^{n\times n}. \tag{8.20}$$

Then the inverse tensor γ^{-1} exists and we define for any matrix-valued function $\Phi \in L_2(\Omega, R^{n\times n})$, analogically to (8.1), the norms

$$|||\Phi|||^2 = \int_\Omega (\gamma\Phi) \odot \Phi \quad \text{and} \quad |||\Phi|||_*^2 = \int_\Omega \left(\gamma^{-1}\Phi\right) \odot \Phi.$$

A function $u \in H_0^2(\Omega)$ is now said to be the weak solution of the problem (8.18), (8.19) if it satisfies the identity

$$\int_\Omega (\gamma\nabla\nabla u) \odot (\nabla\nabla v) = \int_\Omega f v$$

for all test functions $v \in H_0^2(\Omega)$.

Let $\bar{u}$ be a function from $H_0^2(\Omega)$ considered as an approximation of the weak solution u. In [16], no specification of the way $\bar{u}$ has been computed is required, it is just an arbitrary function of the admissible class.

Define the *global functional a posteriori error estimator*

$$\varepsilon_{\mathrm{R}}(\beta, \Phi, \bar{u}) = (1+\beta)|||\gamma\nabla\nabla\bar{u} - \Phi|||_*^2 + \left(1 + \frac{1}{\beta}\right) C_{1\Omega}^2 \|\operatorname{div}\operatorname{Div}\Phi - f\|_0^2,$$

where β is an arbitrary positive real number, Φ an arbitrary symmetric matrix-valued function from $H(\operatorname{div}\operatorname{Div}, \Omega)$, and $C_{1\Omega}$ the constant from the Friedrichs inequality

$$\|w\|_0 \le C_{1\Omega} |||\nabla\nabla w||| \tag{8.21}$$

valid for all $w \in H_0^2(\Omega)$. Then the following theorem holds [16].

Theorem 8.3 *Let $u \in H_0^2(\Omega)$ be the weak solution of the problem (8.18), (8.19) and $\bar{u} \in H_0^2(\Omega)$ an arbitrary function. Then*

$$\left|||\nabla\nabla(\bar{u} - u)|||^2 \le \varepsilon_{\mathrm{R}}(\beta, \Phi, \bar{u}) \right. \tag{8.22}$$

for any symmetric matrix-valued function $\Phi \in H(\operatorname{div}\operatorname{Div}, \Omega)$ and any positive number β.

The proof of the theorem is based on a more general statement proven in [16]. The estimate (8.22) corresponds to the decomposition $\operatorname{div}\operatorname{Div}\Theta = f$, $\Theta = \gamma\nabla\nabla u$ of Eq. (8.18). However, the condition $\operatorname{div}\operatorname{Div}\Theta \in L_2(\Omega)$ is rather demanding.

To avoid possible difficulties of this kind, we can derive another error estimate if we introduce a further global functional error estimator,

$$\begin{aligned}\widetilde{\varepsilon}_{\mathrm{R}}(\beta, \Phi, Y, \bar{u}) = (1+\beta)|||\gamma\nabla\nabla\bar{u} - \Phi|||_*^2 \\ + \frac{1+\beta}{\beta}(C_{1\Omega}\|\operatorname{div} Y - f\|_0 + C_{2\Omega}\|\operatorname{Div}\Phi - Y\|_0)^2,\end{aligned}$$

where β is a positive real number, Φ an arbitrary symmetric matrix-valued function from $H(\operatorname{Div}, \Omega)$, $C_{2\Omega}$ the constant from the Friedrichs inequality

$$\|\nabla w\|_0 \le C_{2\Omega} |||\gamma\nabla\nabla w||| \tag{8.23}$$

valid for all $w \in H_0^2(\Omega)$, and Y an arbitrary vector-valued function from $H(\operatorname{div}, \Omega)$. Then we get the same statement as in Theorem 8.3 but with $\widetilde{\varepsilon}_{\mathrm{R}}(\beta, \Phi, Y, \bar{u})$ on the right-hand part of (8.22) (cf. [16], where the proof is given). The estimate corresponds to the decomposition $\operatorname{div} Y = f$, $\operatorname{Div}\Theta = Y$, $\Theta = \gamma\nabla\nabla u$ of Eq. (8.18).

Theorem 8.3 is equivalent to the statements proven in [13, Sect. 6.6]. Moreover, in [13] the authors use another global functional a posteriori error estimator to prove a lower estimate for the error.

The constants $C_{1\Omega}$ and $C_{2\Omega}$ can be estimated from above by $m^{-1}C_{1\square}$ and $m^{-1}C_{2\square}$, where m is the constant from (8.20), and $C_{1\square}$ and $C_{2\square}$ appear in the Friedrichs inequalities (8.21), (8.23) that hold for any $w \in H_0^2(\Omega)$ on a rectangular domain $\square$ containing Ω [16].

A posteriori error estimates for Eq. (8.18) with other boundary conditions can be derived, too. Instead of $C_{1\Omega}$ and $C_{2\Omega}$ they involve constants appearing in inequalities analogous to (8.21) and (8.23).

The biharmonic equation

$$\Delta^2 u = f \quad \text{in } \Omega$$

is a particular case of Eq. (8.18). Considering it with the Dirichlet boundary condition (8.19) and introducing a particular error estimator, we obtain a statement analogous to Theorem 8.3, see [16].

Consider another Dirichlet problem. Let $\mathrm{d}^2 u$ denote the Hessian matrix of a function $u : \Omega \to R$, $u \in H^2(\Omega)$. Let the matrix-valued function $\Lambda = [\lambda_{ik}]$, $\Lambda : \Omega \times R^{n\times n} \to R^{n\times n}$ be measurable and bounded with respect to the variable $x \in \Omega$ and of class C_2 with respect to the matrix variable $\Theta \in R^{n\times n}$.

Let the domain $\Omega \subset R^n$ have a piecewise C_1 boundary. We consider the fourth-order elliptic problem

$$\operatorname{div}^2 \Lambda\big(x, \mathrm{d}^2 u\big) = f \quad \text{in } \Omega, \tag{8.24}$$

$$u = \frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma \tag{8.25}$$

with $f \in L_2(\Omega)$.

Making proper assumptions on the Jacobian arrays $\Lambda'(x, \Theta)$, we get the existence of Λ^{-1}, the inverse of Λ with respect to $\Theta \in R^{n\times n}$ [11].

The problem (8.24), (8.25) has a unique weak solution $u \in H_0^2(\Omega)$ that satisfies

$$\int_\Omega \Lambda\big(x, \mathrm{d}^2 u\big) \odot \mathrm{d}^2 v - \int_\Omega f v = 0 \quad \text{for all } v \in H_0^2(\Omega).$$

Let $\bar{u}$ be a function from $H_0^2(\Omega)$ considered as an approximation of the weak solution u. In [11], no specification of the way $\bar{u}$ has been computed is required, it is just an arbitrary function of the admissible class.

We measure the error of the approximate solution $\bar{u}$ by a functional $E(\bar{u})$ introduced in [11]. For $\bar{u} \in H_0^2(\Omega)$, an arbitrary matrix-valued function $\Psi \in H(\operatorname{div}^2, \Omega) \cap L_\infty(\Omega, R^{n\times n})$ and an arbitrary scalar-valued function $w \in H_0^2(\Omega)$, define the *global functional a posteriori error estimator* $\varepsilon_{\mathrm{K}}(\Psi, w, \bar{u})$ like in [11]. It contains four generally unknown positive constants. The corresponding statement proven there yields an upper a posteriori error estimate. To avoid the computation of Λ^{-1} we can introduce another global functional a posteriori error estimator and reformulate the above mentioned statement. Moreover, the authors prove in [11] that the global estimator $\varepsilon_{\mathrm{K}}(\Psi, w, \bar{u})$ is sharp for a sufficiently smooth weak solution.

8.5 Conclusion

The quantitative properties of the indicators and estimators cannot be easily assessed and compared analytically. There are, however, analytical error estimators for some classes of problems (see, e.g., [11, 13, 18]) that require as few unknown constants as

possible. The a posteriori estimates with unknown constants, however, are not optimal for the practical computation. They can be efficient if they are asymptotically exact.

The computation of the reference solution is rather time-consuming. Nevertheless, we use reference solutions as robust error indicators with no unknown constants to control the adaptive strategies in the most complex finite element computations.

Acknowledgements This research was supported by the Grant Agency of the Academy of Sciences of the Czech Republic under Grant IAA100190803 and by the Academy of Sciences of the Czech Republic under Research Plan AV0Z10190503 of the Institute of Mathematics.

References

1. Ainsworth M, Oden JT (2000) A posteriori error estimation in finite element analysis. Wiley, New York
2. Babuška I, Rheinboldt WC (1978) Error estimates for adaptive finite element computations. SIAM J Numer Anal 15(4):736–754
3. Babuška I, Rheinboldt WC (1978) A posteriori error estimates for the finite element method. Int J Numer Methods Eng 12(10):1597–1615
4. Babuška I, Strouboulis T (2001) The finite element method and its reliability. Clarendon Press, New York
5. Babuška I, Whiteman JR, Strouboulis T (2011) Finite elements. An introduction to the method and error estimation. Oxford University Press, Oxford
6. Beirão da Veiga L, Niiranen J, Stenberg R (2007) A posteriori error estimates for the Morley plate bending element. Numer Math 106(2):165–179
7. Brezzi F, Raviart PA (1977) Mixed finite element methods for 4th order elliptic equations. In: Miller JJH (ed) Topics in numerical analysis III: proceedings of the royal Irish academy conference on numerical analysis. Academic Press, London, pp 33–56
8. Charbonneau A, Dossou K, Pierre R (1997) A residual-based a posteriori error estimator for the Ciarlet-Raviart formulation of the first biharmonic problem. Numer Methods Partial Differ Equ 13(1):93–111
9. Ciarlet PG (1978) The finite element method for elliptic problems. North-Holland, Amsterdam
10. Ciarlet PG, Raviart P-A (1974) A mixed finite element method for the biharmonic equation. In: de Boor C (ed) Mathematical aspects of finite elements in partial differential equations. Proceedings of a symposium conducted by the mathematics research center, the university of Wisconsin–Madison, April 1–3, 1974. Academic Press, New York, pp 125–145
11. Karátson J, Korotov S (2009) Sharp upper global a posteriori error estimates for nonlinear elliptic variational problems. Appl Math 54(4):297–336
12. Liu K, Qin X (2007) A gradient recovery-based a posteriori error estimators for the Ciarlet-Raviart formulation of the second biharmonic equations. Appl Math Sci 1(21–24):997–1007
13. Neittaanmäki P, Repin S (2004) Reliable methods for computer simulation: error control and a posteriori estimates. Elsevier, Amsterdam
14. Pomeranz SB (1995) A posteriori finite element method error estimates for fourth-order problems. Commun Numer Methods Eng 11(3):213–226
15. Rannacher R (1979) On nonconforming and mixed finite element method for plate bending problems. The linear case. RAIRO Anal Numér 13(4):369–387
16. Repin S (2008) A posteriori estimates for partial differential equations. Walter de Gruyter, Berlin
17. Segeth K (2010) A review of some a posteriori error estimates for adaptive finite element methods. Math Comput Simul 80(8):1589–1600

18. Vejchodský T (2006) Guaranteed and locally computable a posteriori error estimate. IMA J Numer Anal 26(3):525–540
19. Verfürth R (1996) A review of a posteriori error estimation and adaptive mesh-refinement techniques. Wiley-Teubner, Stuttgart
20. Wang M, Zhang W (2008) Local a priori and a posteriori error estimate of TQC9 element for the biharmonic equation. J Comput Math 26(2):196–208

Chapter 9
Upper Bound for the Approximation Error for the Kirchhoff-Love Arch Problem

Olli Mali

Abstract In this paper, a guaranteed and computable upper bound of approximation errors for the Kirchhoff-Love arch problem is derived. In general, it belongs to the class of functional a posteriori error estimates. The derivation method uses purely functional arguments and, therefore, the estimates are valid for any conforming approximation within the energy space. The computational implementation of the upper bound is discussed and demonstrated by a numerical example.

9.1 Introduction

We consider a plane arch that has a constant cross section which is small compared to its length. Following [3], the arch and all related functions are presented in the parametrized form. The $\psi : [0, 1] \to \mathbb{R}^2$ is a smooth parametrized non-self-intersecting curve of the curvilinear abscissa s that defines the shape of the arch. The displacement vector $u = (u_1, u_2)$ and the load vector $f = (f_1, f_2)$ are given on a local basis (a_1, a_2) that varies along the arch, where a_1 is the tangential and a_2 is the normal direction. The angle between the horizontal axis and a_1 is denoted as θ. On both ends of the beam, there are known external loads, normal force N, shear force F, and the bending moment M. The mentioned definitions with positive directions of the external loads are depicted in Fig. 9.1. A more advanced formulation of the arch problem based on the control theory can be found in [12, 23], where regularity requirements for ψ are substantially relaxed.

The curvature of the arch is $c : [0, 1] \to \mathbb{R}$,

$$c(s) := \frac{1}{R(s)} = \frac{\psi_2''(s)\psi_1'(s) - \psi_1''(s)\psi_2'(s)}{(\psi_1'^2 + \psi_2^2)^{\frac{3}{2}}}. \tag{9.1}$$

O. Mali (✉)
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland
e-mail: olli.mali@jyu.fi

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_9,

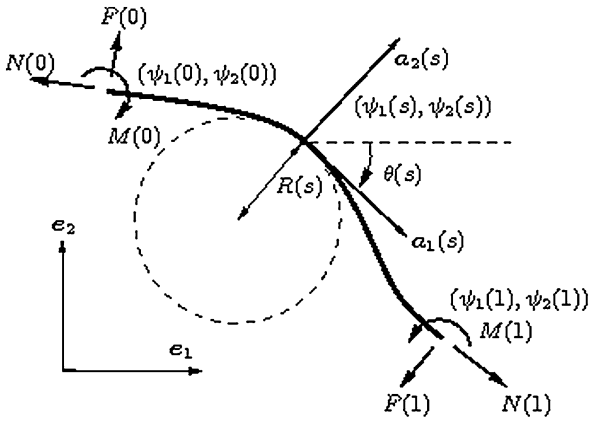


Fig. 9.1 The Kirchhoff-Love arch

The energy functional is

$$J(u) = \frac{1}{2}\int_0^1 \left\{EA\left(u_1' - cu_2\right)^2 + EI\left(cu_1 + u_2'\right)'^2\right\} \mathrm{ds} - \int_0^1 f \cdot u \,\mathrm{ds} - \Big/_0^1 Nu_1 + \Big/_0^1 Fu_2 - \Big/_0^1 Mu_2', \tag{9.2}$$

where E is the material constant (Young's modulus), A is the area of the cross section, and I is the second moment of inertia of the cross section. All these values are strictly positive. We apply the notation

$$\int_a^b f \,\mathrm{ds} = \Big/_a^b F = F(b) - F(a).$$

The minimizer of the energy functional $u \in V_0$ satisfies the integral relation

$$a(u, w) = l(w), \quad \forall\, w \in V_0, \tag{9.3}$$

where

$$a(u, w) = \int_0^1 \begin{bmatrix} EA(u_1' - cu_2) \\ EI(cu_1 + u_2')' \end{bmatrix} \cdot \begin{bmatrix} w_1' - cw_2 \\ (cw_1 + w_2')' \end{bmatrix} \mathrm{ds} \tag{9.4}$$

and

$$l(w) := \int_0^1 f \cdot w \,\mathrm{ds} + \Big/_0^1 Nw_1 - \Big/_0^1 Fw_2 + \Big/_0^1 Mw_2'. \tag{9.5}$$

If u is sufficiently regular, then (9.3) implies the classical equations

$$\begin{cases} -\left(EA\left(u_1' - cu_2\right)\right)' - c\left(EI\left(cu_1 + u_2'\right)'\right)' = f_1, \\ -cEA\left(u_1' - cu_2\right) + \left(EI\left(cu_1 + u_2'\right)'\right)'' = f_2. \end{cases} \tag{9.6}$$

Table 9.1 Boundary conditions of the Kirchhoff-Love arch

Kinematic	Natural
u_1 (tangential disp.)	N (tangential stress)
u_2 (normal disp.)	F (shear force)
u_2' (rotation)	M (bending moment)

The boundary conditions are defined at the end points $s = 0$ and $s = 1$. They are listed as pairs in Table 9.1. Kinematic boundary conditions restrict displacement components or rotation and natural boundary conditions define tangential stress, shear force or bending moment. At both ends of the beam, either a natural or a corresponding kinematic boundary condition has to be defined.

We assume that kinematic boundary conditions are homogeneous. Together with the regularity requirements, kinematic boundary conditions define the space of admissible displacements

$$V_0 := \{v \in V \mid v \text{ satisfies homogeneous kinematic boundary conditions}\}, \quad (9.7)$$

where we denote $V := H^1(0,1) \times H^2(0,1)$. The mentioned assumptions guarantee that in (9.5) either N, V, or M is known or the condition $w \in V_0$ implies the vanishing of the corresponding term.

We note that (9.6) can be decomposed into

$$\begin{cases} EA\big(u_1' - cu_2\big) = p_1, \\ EI\big(cu_1 + u_2'\big)' = p_2 \end{cases} \quad (9.8)$$

and

$$\begin{cases} -p_1' - cp_2' = f_1, \\ -cp_1 + p_2'' = f_2. \end{cases} \quad (9.9)$$

Equations (9.8) are the constitutive relation that states the linear dependence between displacement u and tangential stress p_1 and the bending moment p_2. Henceforth, the vector p will be referred to as the stress vector. Equations (9.9) are the equilibrium condition between the external load f and the stresses p of the beam. At the end points of the beam, stresses must satisfy the natural boundary conditions, namely

$$p_1 + cp_2 = N, \quad p_2 = M, \quad \text{and} \quad p_2' = F. \quad (9.10)$$

The stresses satisfying these relations form the space of admissible stresses,

$$Q_0 := \big\{y \in H^1(0,1) \times H^2(0,1) \mid y \text{ satisfies (9.10)}\big\}. \quad (9.11)$$

The problem is called statically determined (or overdetermined) if for $p = 0$, Eq. (9.8) imply $u = 0$. Then the respective boundary conditions are physically sensible, i.e., the beam without any load "stays still".

We define the following operators:

$$\Lambda u := \begin{bmatrix} u_1' - cu_2 \\ (cu_1 + u_2')' \end{bmatrix}, \qquad \Lambda^* p := \begin{bmatrix} -p_1' - cp_2' \\ -cp_1 + p_2'' \end{bmatrix}, \quad \text{and} \quad \mathscr{A} p := \begin{bmatrix} EA\, p_1 \\ EI\, p_2 \end{bmatrix}. \tag{9.12}$$

Then (9.8) and (9.9) can be written as

$$\mathscr{A} \Lambda u = p \tag{9.13}$$

and

$$\Lambda^* p = f, \tag{9.14}$$

respectively. For the full exposition of the discussed beam theory, see, e.g., [24, 25].

The general existence theory for elliptic equations is well known (see, e.g., [4]). For the existence of the solution of the Kirchhoff-Love arch problem we must show the ellipticity of $a : V_0 \times V_0 \to \mathbb{R}$, which is proved in [3, Theorem 8.1.2, p. 433],

Theorem 9.1 *If the function c is continuously differentiable over the interval I, the bilinear form*

$$a(u, v) = \int_I \left\{ \left(u_1' - cu_2\right)\left(v_1' - cv_2\right) + \left(u_2' + cu_1\right)'(v_2 + cu_1)' \right\} \mathrm{ds}$$

is $H_0^1(I) \times (H^2(I) \cap H_0^1(I))$-elliptic, and, thus, it is a fortiori $H_0^1(I) \times H_0^2(I)$-elliptic.

Theorem 9.1 states that for a statically determinate or overdeterminate beam, there exists a positive constant C such that

$$\int_0^1 \left\{ w_1^2 + w_2^2 + w_1'^2 + w_2'^2 + w_2''^2 \right\} \mathrm{ds} \le C \int_0^1 \left\{ \left(w_1' - cw_2\right)^2 + \left(cw_1 + w_2'\right)'^2 \right\} \mathrm{ds} \tag{9.15}$$

for all $w \in V_0$.

9.2 Error Majorant

The following upper estimate of the deviation from the exact solution (error majorant) for the Kirchhoff-Love arch model was first presented in [10], where also the respective lower estimate is presented. However, in this paper, we discuss the guaranteed estimation of the approximation error, where the majorant is more relevant. Estimates similar (or more complicated) to the one presented here have been derived earlier (see, e.g., [9, 15–18, 20, 21]). Here we derive the majorant with the help of integral identities. The more general variational method for deriving the functional estimates is discussed in [11, 19].

Theorem 9.2 *Let u be a solution of (9.3) and $v \in V_0$. Then*

$$|||u - v|||^2 \le \mathscr{M}_{\oplus}(v, y, \beta), \quad y \in Q_0,\ \beta > 0,$$

where

$$\begin{aligned}\mathscr{M}_{\oplus}(v, y, \beta) := &\left(1 + \frac{1}{\beta}\right)\frac{C}{\alpha}\int_0^1 \left\{\left(f_1 + \left(y_1' + cy_2'\right)\right)^2 + \left(f_2 - \left(cy_1 + y_2''\right)\right)^2\right\} \mathrm{ds} \\ &+ (1+\beta)\int_0^1 \left\{\frac{1}{EA}\left(y_1 - EA\left(v_1' - cv_2\right)\right)^2 \right. \\ &\left. + \frac{1}{EI}\left(y_2 - EI\left(cv_1 + v_2'\right)\right)'^2\right\} \mathrm{ds}. \end{aligned} \tag{9.16}$$

Here C is from (9.15), $\alpha := \min\{EA, EI\}$, and

$$|||w|||^2 := \frac{1}{2}\int_0^1 \left\{EA\left(w_1' - cw_2\right)^2 + EI\left(cw_1 + w_2'\right)'^2\right\} \mathrm{ds}.$$

Proof We note that

$$\begin{aligned}&\int_0^1 \begin{bmatrix} w_1' - cw_2 \\ (cw_1 + w_2')' \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \mathrm{ds} \\ &= \int_0^1 \begin{bmatrix} -y_1' - cy_2' \\ -cy_1 + y_2'' \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \mathrm{ds} + \Big/_0^1 w_1 y_1 + \Big/_0^1 \left(cw_1 + w_2'\right) y_2 - \Big/_0^1 w_2 y_2' \end{aligned} \tag{9.17}$$

for any $w \in H^1(0,1) \times H^2(0,1)$ and $y \in H^1(0,1) \times H^2(0,1)$.

By (9.3) and (9.17) we obtain

$$\begin{aligned} a(u - v, w) = &\int_0^1 f \cdot w \,\mathrm{ds} + \Big/_0^1 Nw_1 - \Big/_0^1 Fw_2 + \Big/_0^1 Mw_2' \\ &- \int_0^1 \begin{bmatrix} EA(v_1' - cv_2) \\ EI(cv_1 + v_2')' \end{bmatrix} \cdot \begin{bmatrix} w_1' - cw_2 \\ (cw_1 + w_2')' \end{bmatrix} \mathrm{ds} \\ &+ \int_0^1 \begin{bmatrix} w_1' - cw_2 \\ (cw_1 + w_2')' \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \mathrm{ds} \\ &- \int_0^1 \begin{bmatrix} -y_1' - cy_2' \\ cy_1 + y_2'' \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \mathrm{ds} - \Big/_0^1 w_1 y_1 \\ &- \Big/_0^1 \left(cw_1 + w_2'\right) y_2 + \Big/_0^1 w_2 y_2'. \end{aligned} \tag{9.18}$$

We rewrite (9.18) in the form

$$a(u - v, w) = I_1 + I_2 + I_3, \tag{9.19}$$

where

$$I_1 = \int_0^1 \begin{bmatrix} f_1 + (y_1' + cy_2') \\ f_2 - (cy_1 + y_2'') \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \mathrm{ds},$$

$$I_2 = \int_0^1 \begin{bmatrix} y_1 - EA(v_1' - cv_2) \\ y_2 - EI(cv_1 + v_2')' \end{bmatrix} \cdot \begin{bmatrix} w_1' - cw_2 \\ (cw_1 + w_2')' \end{bmatrix} \mathrm{ds},$$

$$I_3 = \int_0^1 (N - y_1 - cy_2)w_1 + \int_0^1 \big(-F + y_2'\big)w_2 + \int_0^1 (M - y_2)w_2'.$$

After imposing the boundary conditions, $w \in V_0$ and $y \in Q_0$, I_3 vanishes.

By the Cauchy-Schwartz inequality, we have

$$I_1 \le \left(\int_0^1 \big(f_1 + \big(y_1' + cy_2'\big)\big)^2 + \big(f_2 - \big(cy_1 + y_2''\big)\big)^2 \mathrm{ds} \right)^{\frac{1}{2}} \left(\int_0^1 w_1^2 + w_2^2 \mathrm{ds} \right)^{\frac{1}{2}}.$$

We can estimate the L_2-norm of w from above by the Sobolev norm of $H^1(0,1) \times H^2(0,1)$ and apply (9.15). Then

$$I_1 \le \left(\int_0^1 \big(f_1 + \big(y_1' + cy_2'\big)\big)^2 + \big(f_2 - \big(cy_1 + y_2''\big)\big)^2 \mathrm{ds} \right)^{\frac{1}{2}} \times \frac{\sqrt{C}}{\sqrt{\alpha}} \left(\int_0^1 EA\big(w_1' - cw_2\big)^2 + EI\big(cw_1 + w_2'\big)'^2 \mathrm{ds} \right)^{\frac{1}{2}}, \tag{9.20}$$

where $\alpha = \min\{EA, EI\}$. Now, we apply the Cauchy-Schwartz inequality again,

$$I_2 = \int_0^1 \begin{bmatrix} \frac{1}{\sqrt{EA}}(y_1 - EA(v_1' - cv_2)) \\ \frac{1}{\sqrt{EI}}(y_2 - EI(cv_1 + v_2')') \end{bmatrix} \cdot \begin{bmatrix} \sqrt{EA}(w_1' - cw_2) \\ \sqrt{EI}(cw_1 + w_2')' \end{bmatrix} \mathrm{ds}$$

$$\le \left(\int_0^1 \frac{1}{EA}\big(y_1 - EA\big(v_1' - cv_2\big)\big)^2 + \frac{1}{EI}\big(y_2 - EI\big(cv_1 + v_2'\big)'^2\big) \mathrm{ds} \right)^{\frac{1}{2}} |||w|||. \tag{9.21}$$

We apply (9.21) and (9.20) to (9.19) and set $w = u - v$, then we arrive at

$$|||u - v||| \le \frac{\sqrt{C}}{\sqrt{\alpha}} \left(\int_0^1 \big(f_1 + \big(y_1' + cy_2'\big)\big)^2 + \big(f_2 - \big(cy_1 + y_2''\big)\big)^2 \mathrm{ds} \right)^{\frac{1}{2}} + \left(\int_0^1 \frac{1}{EA}\big(y_1 - EA\big(v_1' - cv_2\big)\big)^2 + \frac{1}{EI}\big(y_2 - EI\big(cv_1 + v_2'\big)'^2\big) \mathrm{ds} \right)^{\frac{1}{2}}. \tag{9.22}$$

For computation purposes it is preferable to have quadratic expressions. Thus we introduce arbitrary $\beta \in \mathbb{R}_+$ and use the Young inequality to obtain

$$\begin{aligned}|||u-v|||^2 \le \left(1+\frac{1}{\beta}\right)\frac{C}{\alpha}\int_0^1 \left\{\left(f_1+\left(y_1'+cy_2'\right)\right)^2+\left(f_2-\left(cy_1+y_2''\right)\right)^2\right\}\mathrm{ds}\\ +(1+\beta)\int_0^1\left\{\frac{1}{EA}\left(y_1-EA\left(v_1'-cv_2\right)\right)^2\right.\\ \left.+\frac{1}{EI}\left(y_2-EI\left(cv_1+v_2'\right)'^2\right)\right\}\mathrm{ds}. \qquad (9.23)\end{aligned}$$

We call the right-hand side the error majorant (or majorant) and denote it by $\mathscr{M}_\oplus(v,y,\beta)$. □

Remark 9.1 Under the definitions (9.12) the majorant has the structure

$$\begin{aligned}\mathscr{M}_\oplus(v,y,\beta) := \left(1+\frac{1}{\beta}\right)\frac{C}{\alpha}\int_0^1 \left|f-\Lambda^* y\right|^2\mathrm{ds}\\ +(1+\beta)\int_0^1 \mathscr{A}^{-1}(y-\mathscr{A}\Lambda v)\cdot(y-\mathscr{A}\Lambda v)\,\mathrm{ds}. \qquad (9.24)\end{aligned}$$

Remark 9.2 Two terms of the error majorant are related to the decomposed form of the classical equations. The first part is the error in the equilibrium condition (9.9). We denote this part by

$$\mathscr{M}_\oplus^{\mathrm{equi}} := \int_0^1 \left|f-\Lambda^* y\right|^2\mathrm{ds}. \qquad (9.25)$$

The second part is the violation of the duality relation (9.8),

$$\mathscr{M}_\oplus^{\mathrm{const}} := \int_0^1 \mathscr{A}^{-1}(\mathscr{A}y-\Lambda v)\cdot(y-\mathscr{A}\Lambda v)\,\mathrm{ds}. \qquad (9.26)$$

If we substitute $y := p$ to the majorant, the second part provides the exact error, and the first part is zero.

9.2.1 Application of the Majorant

The difference between the exact solution u of (9.32) and any approximation $v \in V_0$ can be estimated from above using the majorant as follows:

$$|||u-v|||^2 \le \mathscr{M}_\oplus(v,y,\beta), \quad \forall y \in Q_0,\ \beta>0,$$

where y and β are at our disposal. Often, they are members of some finite dimensional subspaces, $w \in V_0^N \subset V_0$ and $y \in Q^N \subset Q$. The exact selection of basis functions generating V_0^N and Q^N depends on the problem type, computational resources, and the desired accuracy of estimates. For example, they can be piecewise polynomials with highly local support as in a traditional finite element approach.

There are numerous variants of how to select the auxiliary function y. Recall that we would obtain the exact deviation if y is the exact stress $y = p$. In practice, p is not at our disposal. There are two principal ways to select the auxiliary function y:

1. We postprocess the approximate solution (this procedure is denoted by $G : Q \to Q$) to obtain an approximation of the stress,

$$y := G(\mathscr{A}\Lambda v) \approx p.$$

2. We minimize the majorant with respect to the auxiliary function y within some subspace $Q^N \in Q$, i.e., we solve the problem

$$\min_{\substack{y \in Q^N, \\ \beta > 0}} \mathscr{M}_{\oplus}(v, y, \beta).$$

The minimization procedure with respect to y and β can be done iteratively.

If it is necessary to obtain a reasonable upper bound with less computational effort, then the method 1 is preferable. For a more accurate bound the method 2 is recommended. Note that it not only provides an improved upper bound for the error, it also produces a good approximation of the true flux.

We present the method 2 in more detail. The majorant is convex (quadratic) with respect to y. The necessary condition for the minimizer y can be computed as follows:

$$\mathscr{M}_{\oplus}(v, y + t\mu) = \left(1 + \frac{1}{\beta}\right)\frac{C}{\alpha}\int_0^1 \left|f - \Lambda^* y - t\Lambda^*\mu\right|^2 \mathrm{d}s + (1+\beta)\int_0^1 \mathscr{A}^{-1}(y + t\mu - \mathscr{A}\Lambda v)\cdot(y + t\mu - \mathscr{A}\Lambda v)\,\mathrm{d}s.$$

Therefore

$$\frac{\mathrm{d}\mathscr{M}_{\oplus}(v, y + t\mu)}{\mathrm{d}t} = \left(1 + \frac{1}{\beta}\right)\frac{C}{\alpha}2\int_0^1 \left(f - \Lambda^* y - t\Lambda^*\mu\right)\cdot\left(-\Lambda^*\mu\right)\mathrm{d}s + (1+\beta)2\int_0^1 \mathscr{A}^{-1}(y + t\mu - \mathscr{A}\Lambda v)\cdot\mu\,\mathrm{d}s$$

and the condition

$$\left.\frac{\mathrm{d}\mathscr{M}_{\oplus}(v, y + t\mu)}{\mathrm{d}t}\right|_{t=0} = 0$$

reads

$$\left(1+\frac{1}{\beta}\right)\frac{C}{\alpha}\int_0^1 \Lambda^* y\cdot\Lambda^*\mu\,\mathrm{ds}+(1+\beta)\int_0^1 \mathscr{A}^{-1}y\cdot\mu\,\mathrm{ds}$$
$$=\left(1+\frac{1}{\beta}\right)\frac{C}{\alpha}\int_0^1 f\cdot\Lambda^*\mu\,\mathrm{ds}+(1+\beta)\int_0^1 \mathscr{A}\Lambda v\cdot\mu\,\mathrm{ds}. \tag{9.27}$$

Let y belong to a finite dimensional subspace of Q,

$$y\in\operatorname{span}\{\phi^1,\phi^2,\ldots,\phi^N\}=:Q_N\subset Q,$$

i.e.,

$$y=\sum_{i=1}^N \gamma_i\phi^i.$$

Then the condition (9.27) leads to a system of linear equations,

$$\sum_{i=1}^N\gamma_i\left(\left(1+\frac{1}{\beta}\right)\frac{C}{\alpha}\int_0^1\Lambda^*\phi^i\cdot\Lambda^*\phi^j\,\mathrm{ds}+(1+\beta)\int_0^1\mathscr{A}^{-1}\phi^i\cdot\phi^j\,\mathrm{ds}\right)$$
$$=\left(1+\frac{1}{\beta}\right)\frac{C}{\alpha}\int_0^1 f\cdot\Lambda^*\phi^j\,\mathrm{ds}+(1+\beta)\int_0^1\mathscr{A}\Lambda v\cdot\phi^j\,\mathrm{ds},\quad j=\{1,\ldots,N\}.$$

After introducing the matrices

$$\{S_{ij}\}_{i,j=1}^N=\int_0^1\Lambda^*\phi^i\cdot\Lambda^*\phi^j\,\mathrm{ds},\qquad \{K_{ij}\}_{i,j=1}^N=\int_0^1\mathscr{A}^{-1}\phi^i\cdot\phi^j\,\mathrm{ds},$$

and the vectors

$$\{z_j\}_{j=1}^N=\int_0^1 f\cdot\Lambda^*\phi^j\,\mathrm{ds},\qquad \{g_j\}_{j=1}^N=\int_0^1\mathscr{A}\Lambda v\cdot\phi^j\,\mathrm{ds}$$

the system can be written in the matrix form

$$\left(\left(1+\frac{1}{\beta}\right)\frac{C}{\alpha}S+(1+\beta)K\right)\gamma=\left(1+\frac{1}{\beta}\right)\frac{C}{\alpha}z+(1+\beta)g, \tag{9.28}$$

where γ is a vector consisting of the unknown coefficients. The evaluation of the majorant for $y\in Q_N$ can be easily done using the predefined matrices and the coefficient vector γ,

$$\mathscr{M}_\oplus(v,y,\beta)=\left(1+\frac{1}{\beta}\right)\frac{C}{\alpha}\left(\gamma^T S\gamma-2\gamma^T z+\|f\|^2\right)$$
$$+(1+\beta)\left(\gamma^T K\gamma-2\gamma^T g+a(v,v)\right).$$

If the majorant is minimized with respect to a positive scalar β, the minimum value is attained at

$$\beta := \left(\frac{\frac{C}{\alpha} \mathscr{M}_{\oplus}^{\text{equi}}}{\mathscr{M}_{\oplus}^{\text{const}}} \right)^{\frac{1}{2}}. \tag{9.29}$$

These observations motivate Algorithm 9.1.

Algorithm 9.1 Minimization of the majorant

Require: Matrices S and K, and vectors z and g are assembled and constants $\|f\|^2$ and $a(v, v)$ are computed. Set initial β_1.

for $k = 1$ to $I_{\max}$ **do**

Solve:

$$\left(\left(1 + \frac{1}{\beta_k} \right) \frac{C}{\alpha} S + (1 + \beta_k) K \right) \gamma_{k+1} = \left(1 + \frac{1}{\beta_k} \right) \frac{C}{\alpha} z + (1 + \beta_k) g.$$

Compute parts of the majorant:

$$\mathscr{M}_{\oplus}^{\text{equi}} = \gamma_{k+1}^T S \gamma_{k+1} - 2 y_{k+1}^T z + \|f\|^2,$$
$$\mathscr{M}_{\oplus}^{\text{const}} = \gamma_{k+1}^T K \gamma_{k+1} - 2 y_{k+1}^T g + a(v, v).$$

Compute parameter β:

$$\beta_{k+1} = \left(\frac{\frac{C}{\alpha} \mathscr{M}_{\oplus}^{\text{equi}}}{\mathscr{M}_{\oplus}^{\text{const}}} \right)^{\frac{1}{2}}$$

end for

It is not obligatory to solve the equations (9.28) for y exactly. An efficient method for approximating y is the so called multi-grid method proposed in [26]. In general, iterative numerical methods for solving (9.28) are attractive alternatives, since at every iteration step one can compute the value of the majorant and cease all computations after the desired error estimation accuracy is obtained. It is rarely of interest to compute the value of the approximation error as accurately as possible; a reasonable upper bound for it is usually satisfactory. The construction and implementation of the error majorant has been studied for various problems in [2, 5–8].

For the computation of the majorant we need to estimate the constant C_F in the inequality (9.15). In practice, to compute the majorant, we only need to estimate the magnitude of the constant roughly. For example, it can be estimated by solving approximately a generalized eigenvalue problem using the Galerkin approximation: Find eigenpairs (λ_i, v_i), where $v_i \in V_0^N \subset V_0$, such that

$$(\mathscr{A} \Lambda v_i, \Lambda w)_U = \lambda_i (v_i, w)_{\mathscr{V}}, \quad \forall w \in V_0^N. \tag{9.30}$$

The value of the constant is $C_F = \frac{1}{\lambda_{\min}}$, where $\lambda_{\min}$ is the lowest eigenvalue.

Besides the upper bound for the error, the majorant is the basis for various error indicators. It is relevant to distinguish between error indicators and estimators. The main goal of an error indicator is to form an adequate approximation of the error distribution with the lowest possible computational cost. This knowledge is essential for adaptive methods that iteratively enrich the set of basis functions used to compute the approximate solution. The theoretical basis dates back to [13]. In the last decades, error indicators have been intensively studied in numerical analysis and the amount of different methods and implementations is vast and beyond the scope of this paper (see, e.g., [1, 14, 22, 27, 28]).

9.3 Example: Uniformly Curved Beam

We consider a half circular beam

$$\psi(t) = \begin{bmatrix} \cos(\pi t) \\ \sin(\pi t) \end{bmatrix}, \quad t \in [0, 1],$$

where the curvature is $c = 1$. Let both ends of the beam be clamped, i.e., the displacement satisfies the boundary conditions

$$u_1(0) = u_2(0) = u_2'(0) = u_1(1) = u_2(1) = u_2'(1) = 0.$$

We normalize $EA = EI = 1$.

First, we compute an approximation of the constant C in (9.15). The basis that satisfies the boundary conditions can be easily constructed using Fourier-type basis functions. Let

$$w \in V_0^N := \operatorname{span}\left\{ \begin{bmatrix} \sin(k\pi t) \\ 0 \end{bmatrix}, \begin{bmatrix} 1 - \cos(2k\pi t) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 - \cos(2k\pi t) \end{bmatrix} \right\}_{k=1}^{N}. \quad (9.31)$$

Then, $\dim(V_0^N) = 3N$. We approximate C by solving the general eigenvalue problem (9.30) using the Galerkin method. For $N = 10$, we have $C \approx 0.051$. We estimate from above and set $C = 1$.

We introduce a polynomial solution

$$u(t) = \begin{bmatrix} t(t-1) \\ t^2(t-1)^2 \end{bmatrix}, \quad (9.32)$$

which satisfies the kinematic boundary conditions. From (9.8) and (9.9), we have

$$p(t) = \begin{bmatrix} EA(-t^4 + 2t^3 - t^2 + 2t - 1) \\ EI(12t^2 - 10t + 1) \end{bmatrix}$$

and

$$f(t) = \begin{bmatrix} EA(4t^3 - 6t^2 + 2t - 2) + EI(-24t + 10) \\ EA(t^4 - 2t^3 + t^2 - 2t + 1) + 24EI \end{bmatrix}.$$

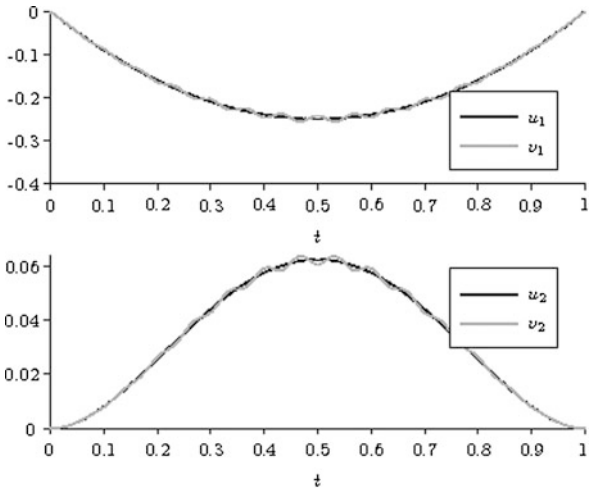


Fig. 9.2 The exact solution u and the "approximation" v

To study the application of a posteriori error estimates, we define an "approximate solution" v of the form

$$v := u + \epsilon\xi \in V_0,$$

where ξ is a known "error" that satisfies the kinematic boundary conditions. We selected

$$\xi := \begin{bmatrix} t(t-1)\cos(30\pi t) \\ t^2(t-1)^2\cos(30\pi t) \end{bmatrix}.$$

We set $\epsilon := 0.02\frac{\|u\|}{\|\xi\|}$ to obtain the following relative error (in the L_2-norm) of 2 %. The "error" in the energy norm is $|||u - v|||^2 = 50.740$. The exact solution and the "approximation" are depicted in Fig. 9.2.

To measure the efficiency of the majorant, we introduce the efficiency index,

$$I^{\oplus}_{\text{eff}} := \frac{\mathscr{M}_{\oplus}}{|||u - v|||^2}. \tag{9.33}$$

Since the majorant is guaranteed,

$$1 \leq I^{\oplus}_{\text{eff}}.$$

Let y in the majorant be defined on a Fourier basis, i.e.,

$$y \in Q_N := \text{span}\left\{ \begin{bmatrix} \sin(k\pi t) \\ 0 \end{bmatrix}, \begin{bmatrix} \cos(k\pi t) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \sin(k\pi t) \end{bmatrix}, \begin{bmatrix} 0 \\ \cos(k\pi t) \end{bmatrix} \right\}_{k=1}^{N}.$$

Note that $\dim(Q_N) = 4N$. We minimize the majorant with respect to $y \in Q_N$ following Algorithm 9.1. Regardless of the dimension N, the iteration converged in six

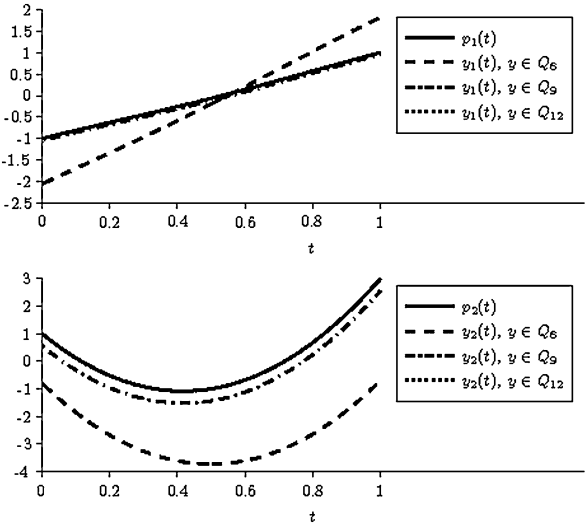


Fig. 9.3 The approximate y obtained through Algorithm 9.1, where the dimension of y varies, compared to the exact stress p

Table 9.2 An efficiency index of the majorant with different values of N

N	4	6	8	9	10	11	12
$I_{\text{eff}}^{\oplus}$	1.1853	1.1509	1.0204	1.0048	1.0011	1.0002	1. 0001
$\mathscr{M}_{\oplus}$	60.143	58.398	51.774	50.985	50.7949	50.752	50.742
$\mathscr{M}_{\oplus}^{\text{equi}}$	0.01320	0.00229	0.00485	0.00029	0.00002	7.2×10^{-7}	3.3×10^{-8}
$\mathscr{M}_{\oplus}^{\text{const}}$	58.374	57.658	50.776	50.742	50.740	50.740	50.740

Table 9.3 An efficiency index of the majorant with different values of N and the constant C

N		4	6	8	9	10	11	12
$C = 1$	$I_{\text{eff}}^{\oplus}$	1.1853	1.1509	1.0204	1.0048	1.0011	1.0002	1. 0001
$C = 10$	$I_{\text{eff}}^{\oplus}$	1.2587	1.1601	1.0601	1.0151	1.0034	1.0008	1.0002
$C = 100$	$I_{\text{eff}}^{\oplus}$	1.4992	1.1747	1.1413	1.0453	1.0108	1.0024	1.0005
$C = 1000$	$I_{\text{eff}}^{\oplus}$	2.3947	1.2026	1.1592	1.1187	1.0329	1.0075	1.0016

steps. In Fig. 9.3, we have depicted y and the exact stress p. Clearly, y approaches p as N increases. For $N = 12$ the difference between the curves is no longer visible.

In Table 9.2, we can observe how the majorant improves as N increases. The efficiency index tends to one as the majorant approaches the exact deviation, and the equilibrium part of the majorant tends to zero as the constitutive part approaches the exact deviation error.

Since the constant C in (9.15) multiplies $\mathscr{M}_{\oplus}^{\text{equi}}$ which tends to zero, even a substantial overestimation of C does not seriously affect the efficiency of the majorant.

In Table 9.3, we show the efficiency index obtained by different values for the constant C.

Acknowledgements The author wants to congratulate Prof. P. Neittaanmäki for his 60th birthday and express his gratitude to him and Prof. S. Repin for their support.

References

1. Ainsworth M, Oden JT (2000) A posteriori error estimation in finite element analysis. Wiley, New York
2. Anjam I, Mali O, Muzalevsky A, Neittaanmäki P, Repin S (2009) A posteriori error estimates for a Maxwell type problem. Russ J Numer Anal Math Model 24(5):395–408
3. Ciarlet PG (1978) The finite element method for elliptic problems. Studies in Mathematics and its applications, vol 4. North-Holland, Amsterdam
4. Ekeland I, Temam R (1976) Convex analysis and variational problems. Studies in mathematics and its applications, vol 1. North-Holland, Amsterdam
5. Frolov M (2004) Reliable control over approximation errors by functional type a posteriori estimates. PhD thesis, University of Jyväskylä
6. Frolov M, Neittaanmäki P, Repin S (2004) On computational properties of a posteriori error estimates based upon the method of duality error majorants. In: Numerical mathematics and advanced applications (ENUMATH 2003, Prague). Springer, Berlin, pp 346–357
7. Gorshkova E (2007) A posteriori error estimates and adaptive methods for incompressible viscous flow problems. PhD thesis, University of Jyväskylä
8. Gorshkova E, Mahalov A, Neittaanmäki P, Repin S (2007) A posteriori error estimates for viscous flow problems with rotation. J Math Sci (NY) 142(1):1749–1762
9. Hannukainen A (2008) Functional type a posteriori error estimates for Maxwell's equations. In: Numerical mathematics and advanced applications (ENUMATH 2007). Springer, Berlin, pp 41–48
10. Mali O (2009) A posteriori error estimates for the Kirchhoff-Love arch model. In: Mäkinen R, Neittaanmäki P, Tuovinen T, Valpe K (eds) Proceedings of the 10th finnish mechanics days, number A 1/2009. Reports of the Department of Mathematical Information Technology, Series A, Collections, pp 315–323
11. Neittaanmäki P, Repin S (2004) Reliable methods for computer simulation. Error control and a posteriori estimates. Elsevier, Amsterdam
12. Neittaanmäki P, Sprekels J, Tiba D (2006) Optimization of elliptic systems. Theory and applications. Springer Monographs in Mathematics. Springer, New York
13. Prager W, Synge JL (1947) Approximations in elasticity based on the concept of function space. Q Appl Math 5:241–269
14. Rannacher R (2000) The dual-weighted-residual method for error control and mesh adaptation in finite element methods. In: Whiteman J (ed) The mathematics of finite elements and applications X, MAFELAP 1999 (Uxbridge). Elsevier, Oxford, pp 97–116
15. Repin S (1997) A posteriori error estimates for approximate solutions of variational problems with power growth functionals. J Math Sci (NY) 101(5):3531–3538
16. Repin S (2000) A posteriori error estimation for variational problems with uniformly convex functionals. Math Comput 69(230):481–500
17. Repin S (2003) A posteriori error estimates taking into account indeterminacy of the problem data. Russ J Numer Anal Math Model 18(6):507–519
18. Repin S (2003) Two-sided estimates of deviation from exact solutions of uniformly elliptic equations. In: Proceedings of the St. Petersburg mathematical society, Vol. IX. Amer. Math. Soc. Transl. Ser. 2, vol 209. AMS, Providence, pp 143–171

19. Repin S (2008) A posteriori estimates for partial differential equations. Walter de Gruyter, Berlin
20. Repin S, Sauter S (2006) Functional a posteriori estimates for the reaction-diffusion problem. CR Math Acad Sci Paris 343(5):349–354
21. Repin S, Sauter S, Smolianski A (2003) A posteriori error estimation for the Dirichlet problem with account of the error in the approximation of boundary conditions. Computing 70(3):205–233
22. Stenberg R (1990) Error analysis of some finite element methods for the Stokes problem. Math Comput 54(190):495–508
23. Tiba D, Vodák R (2005) A general asymptotic model for Lipschitzian curved rods. Adv Math Sci Appl 15(1):137–198
24. Timoshenko S, Goodier JN (1951) Theory of elasticity, 2nd edn. McGraw-Hill, New York
25. Timoshenko S, Young DH (1945) Theory of structures. McGraw-Hill, New York
26. Valdman J (2009) Minimization of functional majorant in a posteriori error analysis based on $H(\mathrm{div})$ multigrid-preconditioned CG method. Adv Numer Anal, 2009. Art ID 164519
27. Verfurth R (1996) A review of a posteriori error estimation and adaptive mesh-refinement techniques. Wiley-Teubner, New York
28. Zienkiewicz OC, Zhu JZ (1987) A simple error estimator and adaptive procedure for practical engineering analysis. Int J Numer Methods Eng 24(2):337–357

Chapter 10
Guaranteed Error Bounds for a Class of Picard-Lindelöf Iteration Methods

Svetlana Matculevich, Pekka Neittaanmäki, and Sergey Repin

Abstract We present a new version of the Picard-Lindelöf method for ordinary differential equations (ODEs) supplied with guaranteed and explicitly computable upper bounds of an approximation error. The upper bounds are based on the Ostrowski estimates and the Banach fixed point theorem for contractive operators. The estimates derived in the paper take into account interpolation and integration errors and, therefore, provide objective information on the accuracy of computed approximations.

10.1 Introduction

In this paper, we discuss a new version of the Picard-Lindelöf method for solving the Cauchy problem

$$\frac{du}{dt} = \varphi\big(u(t), t\big), \qquad u(t_0) = u_0, \tag{10.1}$$

where the solution $u(t)$ (which may be a scalar or vector function) must be found on the interval $[t_0, t_K]$.

Existence and uniqueness of the solutions follow from the Picard-Lindelöf theorem and the Picard existence theorem or from the Cauchy-Lipschitz theorem (see [1, pp. 1–15], [3]).

S. Matculevich (✉) · P. Neittaanmäki · S. Repin
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland
e-mail: svmatsul@student.jyu.fi

P. Neittaanmäki
e-mail: pekka.neittaanmaki@mit.jyu.fi

S. Repin
V.A. Steklov Institute of Mathematics in St. Petersburg, Fontanka 27, 191024, St. Petersburg, Russia
e-mail: repin@pdmi.ras.ru

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_10,

The problem (10.1) can be numerically solved by various well-known methods (e.g., the methods of Runge-Kutta and Adams). Typically, the methods are furnished by a priori asymptotic estimates which show theoretical properties of the iteration algorithm. However, these estimates may have mainly a qualitative meaning and do not provide all necessary information about the exact error bounds for particular numerical approximation. This is the goal of a posteriori error estimation methods. We deduce such type of estimates and suggest a version of the Picard-Lindelöf method as a tool for constructing a fully reliable approximation of (10.1).

The Picard-Lindelöf iteration is one of the well-known numerical methods for ODEs. Furthermore, it can be used not only for ODEs but for t-dependent algebraic and functional equations (see, e.g., [5, 6]). It was shown that the speed of convergence is quite independent of the step sizes. Numerical methods based on Picard-Lindelöf iterations for dynamical processes (the so-called waveform relaxation in the context of electrical networks) are discussed in [2].

The approach discussed in this paper is based on two-sided a posteriori estimates derived by Ostrowski [7] (see also systematic exposition presented in the books [4, 8]). The algorithm includes natural adaptation of the integration step and provides guaranteed bounds for the accuracy on the time interval $[t_0, t_K]$.

In Sect. 10.2, we present the main idea of the Picard-Lindelöf method and obtain the conditions which not only provide convergence of the method but also allow applying a posteriori error estimates. However, these estimates cannot be directly used. In practice computations based on the Picard-Lindelöf method we must take into account interpolation and integration errors. This analysis is done in Sect. 10.3. It leads to error bounds, derived in Sect. 10.4, which include the interpolation and integration errors. The structure of the algorithm is exposed in Sect. 10.5, where results of numerical tests are presented.

10.2 The Picard-Lindelöf Method

Assume that the function $\varphi(\xi(t), t)$ (which is allowed to be a vector-valued function) in (10.1) is continuous with respect to both variables in terms of the continuous norm

$$\|u\|_{C([t_k,t_{k+1}])} := \max_{t\in[t_k,t_{k+1}]} \big|u(t)\big|$$

and satisfies the Lipschitz condition in the form

$$\begin{aligned} &\big\|\varphi(u_2,t_2)-\varphi(u_1,t_1)\big\|_{C([t_1,t_2])} \le \mathrm{L}_1\|u_2-u_1\|_{C([t_1,t_2])} + \mathrm{L}_2|t_2-t_1|, \\ &\quad \forall (u_1,t_1),(u_2,t_2)\in Q, \end{aligned} \tag{10.2}$$

where L_1, L_2 are Lipschitz constants, and

$$Q := \big\{(\xi,t)\mid \xi\in \mathrm{U},\ t_0\le t\le t_N\big\}. \tag{10.3}$$

U is the set of possible values of u which comes from an a priori analysis of the problem. (It is clear that $u_0 \in \mathrm{U}$.)

In the Picard-Lindelöf method, we represent the differential equation in the integral form

$$u(t) = \int_{t_0}^{t} \varphi\big(u(s), s\big)\, ds + u_0. \tag{10.4}$$

Now, the exact solution is a fixed point of (10.4), which can be found by the iteration method

$$u_j(t) = \int_{t_0}^{t} \varphi\big(u_{j-1}(s), s\big)\, ds + u_0.$$

We write in the form $u_j = \mathscr{T} u_{j-1} + u_0$, where $\mathscr{T} : X \to X$ is the integral operator.

It is easy to show that the operator

$$\mathscr{T} u := \int_{t_k}^{t} \varphi\big(u(\tau), \tau\big) d\tau + u_{0,k}$$

is q-contractive on $I_k = [t_k, t_{k+1}]$, where I_k is a subinterval of the mesh $\mathscr{F}_K = \bigcup_{k=0}^{K-1} [t_k, t_{k+1}]$ defined on the interval $[t_0, t_K]$, with respect to the norm $\|u\|_{C(I_k)}$, if the condition

$$q := \mathrm{L}_1 (t_{k+1} - t_k) < 1 \tag{10.5}$$

is provided.

Therefore, if the interval $[t_{k+1}, t_k]$ is small enough, then the solution can be found by the iteration procedure. In the next sections, we call this method the Adaptive Picard-Lindelöf (APL) method.

10.3 Application of the Ostrowski Estimates

For the considered problem, the Ostrowski estimate reads as follows:

Theorem 10.1 ([7]) *Assume that* (10.5) *is satisfied on* $I_k := [t_k, t_{k+1}]$. *Then, the following estimate holds*:

$$M_j^{\ominus} := \frac{1}{1+q} \|u_j - u_{j+1}\|_{C(I_k)} \le \|u - u_j\|_{C(I_k)} \le \frac{q}{1-q} \|u_j - u_{j-1}\|_{C(I_k)} =: M_j^{\oplus}. \tag{10.6}$$

Remark 10.1 It is possible to derive more accurate error bounds for $\|u - u_j\|_{C(I_k)}$ by using additional elements of the sequence $\{u_j\}_{j=1}^{\infty}$ that have indexes greater than j:

$$\|u - u_j\|_{C(I_k)} \le M_j^{\oplus, p} := \frac{1}{1-q^p} \|u_j - u_{j+p}\|_{C(I_k)}.$$

By the mathematical induction method it can be proved that the optimal form of the majorant and minorant based on P correspondent elements of the sequence are as follows:

$$M_j^{\ominus,P} := \sup_{p=1,\dots,P}\left\{\frac{1}{1+q^p}\|u_j - u_{j+p}\|_{C(I_k)}\right\},$$

$$M_j^{\oplus,P} := \inf_{p=1,\dots,P}\left\{\frac{1}{1-q^p}\|u_j - u_{j+p}\|_{C(I_k)}\right\}.$$

However, the estimates (10.6) cannot be directly used because numerical approximations include interpolation and integration errors, which must be taken into account by fully reliable schemes.

Let us discuss this issue within the paradigm of a single (e.g., the first) step of the APL:

$$u_1(t) = \int_{t_0}^{t} \varphi\big(u_0(\tau),\tau\big)d\tau, \quad t \in [t_0, t_1],$$

where u_0 is the initial approximation defined as a piecewise affine function on the mesh $\Omega_{S_k} = \bigcup_{s=0}^{S_k-1}[z_s, z_{s+1}]$ on the interval $[t_0, t_1]$.

If $q < 1$ and u_1 is computed exactly, then

$$\big\|u_1(t) - u(t)\big\|_{C([t_0,t_1])} \le \frac{q}{1-q}\big\|u_1(t) - u_0(t)\big\|_{C([t_0,t_1])}. \tag{10.7}$$

However, in general, u_1 is approximated by a piecewise affine continuous function

$$\bar{u}_1(t) = \pi u_1 \in CP^1\big([z_s, z_{s+1}]\big), \quad s = 0,\dots,S_k - 1,$$

where π is the projection operator $\pi : C \to CP^1([t_0, t_1])$ satisfying the relation $\pi\, u(z_s) = \bar{u}(z_s)$. Thus, on the right-hand side of (10.7) we can estimate as follows:

$$\big\|u_1(t) - u_0(t)\big\|_{C([t_0,t_1])} \le \big\|\bar{u}_1(t) - u_0(t)\big\|_{C([t_0,t_1])} + \big\|\bar{u}_1(t) - u_1(t)\big\|_{C([t_0,t_1])}. \tag{10.8}$$

Here $\|\bar{u}_1(t) - u_1(t)\|_{C([t_0,t_1])} = \|\bar{e}_1\|_{C([t_0,t_1])}$ is an interpolation error. In general, this term is unknown, but we can estimate it using an interpolation error estimate.

Numerical integration generates other errors which must be taken into account. Indeed, the values $\bar{u}(z_s)$, $s = 0,\dots,S_k$, cannot be found exactly. Hence, at every node z_s instead of $\bar{u}_1(z_s)$ we have $\widehat{u}_1(z_s)$. Now, (10.8) implies

$$\begin{aligned}\big\|u_1(t) - u_0(t)\big\|_{C([t_0,t_1])} \le{}& \big\|\widehat{u}_1(t) - u_0(t)\big\|_{C([t_0,t_1])} + \big\|\widehat{u}_1(t) - \bar{u}_1(t)\big\|_{C([t_0,t_1])} \\ &+ \big\|\bar{u}_1(t) - u_1(t)\big\|_{C([t_0,t_1])},\end{aligned} \tag{10.9}$$

where $\|\widehat{u}_1(t) - \bar{u}_1(t)\|_{C([t_0,t_1])} = \|\widehat{e}_1\|_{C([t_0,t_1])}$ is the integration error.

10.4 Estimates of Interpolation and Integration Errors

10.4.1 Interpolation Error

We study the difference between u_1 and $\bar{u}_1$, where $\bar{u}_1$ is the linear interpolant of u_1 defined at the points $\{z_s\}_{s=0}^{S_k}$:

$$u_1(z_s) = \bar{u}_1(z_s) = \int_0^{z_s} \varphi\big(u_0(t), t\big)\, dt.$$

For all $z \in [z_s, z_{s+1}]$,

$$\bar{u}_1(z) = u_1(z_s) + \frac{u_1(z_{s+1}) - u_1(z_s)}{\Delta_s}(z - z_s).$$

Then,

$$\begin{aligned}
\bar{e} &= \bar{u}_1(z) - u_1(z) \\
&= \left[\int_0^{z_s} \varphi\big(u_0(t), t\big)\, dt + \frac{\int_{z_s}^{z_{s+1}} \varphi(u_0(t), t) dt}{\Delta_s}(z - z_s)\right] - \int_0^{z} \varphi\big(u_0(t), t\big)\, dt \\
&= \frac{z - z_s}{\Delta_s} \int_{z_s}^{z_{s+1}} \varphi\big(u_0(t), t\big) dt - \int_{z_s}^{z} \varphi\big(u_0(t), t\big)\, dt. \qquad (10.10)
\end{aligned}$$

Taking into account that u_0 is affinely interpolated, consider the last integral on the right-hand side of (10.10)

$$\int_{z_s}^{z} \varphi\big(u_0(t), t\big)\, dt = \int_{z_s}^{z} \varphi\left(u_{0,s} + \frac{u_{0,s+1} - u_{0,s}}{\Delta_s}(t - z_s), t\right) dt. \qquad (10.11)$$

Define

$$\lambda = \frac{t - z_s}{\Delta_s} = \frac{t - z_s}{z_{s+1} - z_s}, \qquad (10.12)$$

where z_s and z_{s+1} are nodes of the mesh defined in Sect. 10.3. Substitute $t = z_s + (z_{s+1} - z_s)\lambda$ to $\varphi(u_0(t), t)$

$$\begin{aligned}
&\varphi\left(u_{0,s} + \frac{u_{0,s+1} - u_{0,s}}{\Delta_s}(t - z_s), t\right) \\
&\quad = \varphi\big(u_{0,s} + (u_{0,s+1} - u_{0,s})\lambda, z_s + \lambda(z_{s+1} - z_s)\big) \\
&\quad = \varphi\big(\lambda u_{0,s+1} + (1 - \lambda)u_{0,s}, \lambda z_{s+1} + (1 - \lambda)z_s\big).
\end{aligned}$$

Let

$$\widetilde{\varphi}_{[s,s+1]} := \varphi_s + \frac{\varphi_{s+1} - \varphi_s}{\Delta_s}(t - z_s), \qquad (10.13)$$

where $\varphi_s = \varphi(u_{0,s}, z_s)$ and $\varphi_{s+1} = \varphi(u_{0,s+1}, z_{s+1})$. Using (10.12), we rewrite (10.13)

$$\widetilde{\varphi}_{[s,s+1]} = \varphi_s + (\varphi_{s+1} - \varphi_s)\lambda = \lambda\varphi_{s+1} + (1-\lambda)\varphi_s. \tag{10.14}$$

Thus, we can derive the following estimate with the help of (10.14) and (10.2):

$$\begin{aligned}
&\left|\varphi\left(u_{0,s} + \frac{u_{0,s+1} - u_{0,s}}{\Delta_s}(t - z_s), t\right) - \widetilde{\varphi}_{[s,s+1]}\right| \\
&\quad \le \left|\varphi\big(\lambda u_{0,s+1} + (1-\lambda)u_{0,s},\ \lambda z_{s+1} + (1-\lambda)z_s\big) - \lambda\varphi_{s+1} + (1-\lambda)\varphi_s\right| \\
&\quad \le (1-\lambda)\big[\mathrm{L}_{1,s}\left|\lambda u_{0,s+1} + (1-\lambda)u_{0,s} - u_{0,s}\right| \\
&\qquad + \mathrm{L}_{2,s}\left|\lambda z_{s+1} + (1-\lambda)z_s - z_s\right|\big] \\
&\qquad + \lambda\big[\mathrm{L}_{1,s}\left|\lambda u_{0,s+1} + (1-\lambda)u_{0,s} - u_{0,s+1}\right| \\
&\qquad + \mathrm{L}_{2,s}\left|\lambda z_{s+1} + (1-\lambda)z_s - z_{s+1}\right|\big] \\
&\quad \le 2\lambda(1-\lambda)\big[\mathrm{L}_{1,s}|u_{0,s+1} - u_{0,s}| + \mathrm{L}_{2,s}|z_{s+1} - z_s|\big] \\
&\quad \le 2\frac{(z_{s+1} - t)(t - z_s)}{\Delta_s^2}\big[\mathrm{L}_{1,s}|u_{0,s+1} - u_{0,s}| + \mathrm{L}_{2,s}\Delta_s\big].
\end{aligned} \tag{10.15}$$

We decompose (10.11)

$$\begin{aligned}
&\int_{z_s}^{z} \varphi\big(u_0(t), t\big)dt \\
&\quad = \int_{z_s}^{z} \widetilde{\varphi}_{[s,s+1]}(t)dt + \int_{z_s}^{z}\left[\varphi\left(u_{0,s} + \frac{u_{0,s+1} - u_{0,s}}{\Delta_s}(t - z_s), t\right) - \widetilde{\varphi}_{[s,s+1]}\right]dt.
\end{aligned} \tag{10.16}$$

Let us denote the first integral on the right-hand side of (10.16) by $\widetilde{i}_s(z)$. Then,

$$\widetilde{i}_s(z) := \int_{z_s}^{z}\left(\varphi_s + \frac{\varphi_{s+1} - \varphi_s}{\Delta_s}(t - z_s)\right)dt = (z - z_s)\left[\varphi_s + \frac{\varphi_{s+1} - \varphi_s}{2\Delta_s}(z - z_s)\right]. \tag{10.17}$$

The second integral on the right-hand side of (10.16) is estimated with the help of (10.15):

$$\begin{aligned}
&\int_{z_s}^{z}\left|\varphi\left(u_{0,s} + \frac{u_{0,s+1} - u_{0,s}}{\Delta_s}(t - z_s), t\right) - \widetilde{\varphi}_{[s,s+1]}\right|dt \\
&\quad \le \frac{2[\mathrm{L}_{1,s}|u_{0,s+1} - u_{0,s}| + \mathrm{L}_{2,s}\Delta_s]}{\Delta_s^2}\int_{z_s}^{z}(t - z_s)(z_{s+1} - t)dt \\
&\quad = \frac{2[\mathrm{L}_{1,s}|u_{0,s+1} - u_{0,s}| + \mathrm{L}_{2,s}\Delta_s]}{\Delta_s^2}\int_{z_s}^{z}(t - z_s)(z_s + \Delta_s - t)dt
\end{aligned}$$

$$= \frac{2[\mathrm{L}_{1,s}|u_{0,s+1}-u_{0,s}|+\mathrm{L}_{2,s}\Delta_s]}{\Delta_s^2}(z-z_s)^2\left[\frac{\Delta_s}{2}-\frac{z-z_s}{3}\right]$$
$$= \frac{[\mathrm{L}_{1,s}|u_{0,s+1}-u_{0,s}|+\mathrm{L}_{2,s}\Delta_s]}{3\Delta_s^2}(z-z_s)^2(2z_s+3\Delta_s-2z).$$

Since

$$\max_{z\in[z_s,z_{s+1}]}(z-z_s)^2(2z_s+3\Delta_s-2z)=\Delta_s^3,$$

we find that

$$\begin{aligned}\int_{z_s}^{z}\left|\varphi\left(u_{0,s}+\frac{u_{0,s+1}-u_{0,s}}{\Delta_s}(t-z_s),t\right)-\widetilde{\varphi}_{[s,s+1]}\right|dt&\\ \le \frac{[\mathrm{L}_{1,s}|u_{0,s+1}-u_{0,s}|+\mathrm{L}_{2,s}\Delta_s]\Delta_s^3}{3\Delta_s^2}&\\ = \frac{[\mathrm{L}_{1,s}|u_{0,s+1}-u_{0,s}|+\mathrm{L}_{2,s}\Delta_s]\Delta_s}{3}.&\end{aligned}\tag{10.18}$$

We represent the interpolation error (10.10) using (10.17),

$$\begin{aligned}\bar{u}_1(z)-u_1(z)&=\frac{z-z_s}{\Delta_s}\int_{z_s}^{z_{s+1}}\varphi\big(u_0(t),t\big)dt-\int_{z_s}^{z}\varphi\big(u_0(t),t\big)dt\\ &=\frac{z-z_s}{\Delta_s}\widetilde{i}_s(z_{s+1})-\widetilde{i}_s(z)+\varepsilon_1(z)+\varepsilon_2(z),\end{aligned}$$

where

$$\varepsilon_1=\int_{z_s}^{z_{s+1}}\left|\varphi\left(u_{0,s}+\frac{u_{0,s+1}-u_{0,s}}{\Delta_s}(t-z_s),t\right)-\widetilde{\varphi}_{[s,s+1]}\right|dt,$$
$$\varepsilon_2=\int_{z_s}^{z}\left|\varphi\left(u_{0,s}+\frac{u_{0,s+1}-u_{0,s}}{\Delta_s}(t-z_s),t\right)-\widetilde{\varphi}_{[s,s+1]}\right|dt.$$

Thus, we estimate the interpolation error as follows:

$$\begin{aligned}\bar{e}&=\big\|\bar{u}_1(z)-u_1(z)\big\|_{C([z_s,z_{s+1}])}\\ &\le\max_{z\in[z_s,z_{s+1}]}\left|\frac{z-z_s}{\Delta_s}\widetilde{i}_s(z_{s+1})-\widetilde{i}_s(z)\right|+\max_{z\in[z_s,z_{s+1}]}\big|\varepsilon_1(z)+\varepsilon_2(z)\big|.\end{aligned}\tag{10.19}$$

For the first term on the right hand side of (10.19) we have (see (10.17))

$$\begin{aligned}\max_{z\in[z_s,z_{s+1}]}\left|\frac{z-z_s}{\Delta_s}\widetilde{i}_s(z_{s+1})-\widetilde{i}_s(z)\right|dt&\le\frac{|\varphi_{s+1}-\varphi_s|}{2\Delta_s}\max_{z\in[z_s,z_{s+1}]}\big|(z-z_s)(z_{s+1}-z)\big|\\ &\le\frac{|\varphi_{s+1}-\varphi_s|}{2\Delta_s}\frac{\Delta_s^2}{4}=\frac{1}{8}|\varphi_{s+1}-\varphi_s|\Delta_s.\end{aligned}$$

For the second term, we have (see (10.18))

$$\max_{z\in[z_s,z_{s+1}]}\left|\varepsilon_1(z)+\varepsilon_2(z)\right| \le 2\frac{\Delta_s[\mathrm{L}_{1,s}|u_{0,s+1}-u_{0,s}|+\mathrm{L}_{2,s}\Delta_s]}{3}.$$

Hence, the overall estimate of the interpolation error has the form

$$\left\|\bar{u}_1(z)-u_1(z)\right\|_{C([z_s,z_{s+1}])} \le \frac{\varphi_{s+1}-\varphi_s}{8}\Delta_s+\frac{2}{3}\Delta_s\left[\mathrm{L}_{1,s}|u_{0,s+1}-u_{0,s}|+\mathrm{L}_{2,s}\ \Delta_s\right]. \tag{10.20}$$

10.4.2 Integration Error

The interpolation error estimate (10.20) does not account for the fact that computations of the integral are performed approximately. It is not difficult to evaluate the integration errors by noting that for a Lipschitz function $f(t)$ the error encompassed in the simplest trapezoidal quadrature formula

$$\int_{t_0}^{t_1} f(t)dt \simeq \frac{f(t_0)+f(t_1)}{2}(t_1-t_0)$$

can be estimated as follows:

$$e_{int} \le \frac{L}{4}(t_1-t_0)^2-\frac{1}{4L}\left[f(t_1)-f(t_0)\right]^2.$$

Then, it is not difficult to show that the integration error can be estimated as

$$\left\|\widehat{u}_1(t)-\bar{u}_1(t)\right\|_{C([z_s,z_{s+1}])} \le \frac{\mathrm{L}_s}{4}\Delta_s^2-\frac{1}{4\mathrm{L}_s}[\varphi_{s+1}-\varphi_s]^2,$$

where $\mathrm{L}_s=\mathrm{L}_{1,s}\,l_s+\mathrm{L}_{2,s}$. (Here, l_s is the slope of the piecewise function on every interval $[z_s, z_{s+1}]$, $s=0,\ldots,S_k-1$.)

10.4.3 Guaranteed Error Bounds for Picard-Lindelöf Method

Thus, on every subinterval $[z_s, z_{s+1}]$ the interpolation error can be estimated with the help of (10.20). Then, for whole interval $[t_0, t_1] := \bigcup_{s=0}^{S_k-1}[z_s, z_{s+1}]$ the interpolation error estimate is the following:

$$\begin{aligned}&\left\|\bar{u}_1(t)-u_1(t)\right\|_{C([t_0,t_1])}\\ &\quad\le \sum_{s=0,\ldots,S_k-1}\frac{\varphi_{s+1}-\varphi_s}{8}\Delta_s+\frac{2}{3}\left[\mathrm{L}_{1,s}|u_{0,s+1}-u_{0,s}|+\mathrm{L}_{2,s}\Delta_s\right]\Delta_s.\end{aligned}$$

Analogously, for the integration error

$$\left\|\bar{u}_1(t)-\widehat{u}_1(t)\right\|_{C([t_0,t_1])} \le \sum_{s=0,\dots,S_k-1} \frac{\mathrm{L}_s}{2}\Delta_s^2 - \frac{1}{2\mathrm{L}_s}[\varphi_{s+1}-\varphi_s]^2.$$

Then, the inequality (10.9) implies the estimate

$$\begin{aligned}
&\left\|u_1(t)-u_0(t)\right\|_{C([t_0,t_1])}\\
&\quad\le \left\|\widehat{u}_1(t)-u_0(t)\right\|_{C([t_0,t_1])}\\
&\qquad+\sum_{s=0,\dots,S_k-1}\left(\frac{\varphi_{s+1}-\varphi_s}{8}\Delta_s+\frac{2}{3}\Delta_s\left[\mathrm{L}_{1,s}|u_{0,s+1}-u_{0,s}|+\mathrm{L}_{2,s}\Delta_s\right]\right)\\
&\qquad+\sum_{s=0,\dots,S_k-1}\left(\frac{\mathrm{L}_s}{2}\Delta_s^2-\frac{1}{2\mathrm{L}_s}[\varphi_{s+1}-\varphi_s]^2\right).
\end{aligned}$$

After j steps of the iterations we obtain

$$\begin{aligned}
\left\|u_{j+1}(t)-u_j(t)\right\|_{C([t_0,t_1])} &\le \mathrm{M}_{j+1}^{\oplus,1}(\widehat{u}_j)\\
&:= \left\|\widehat{u}_{j+1}(t)-\widehat{u}_j(t)\right\|_{C([t_0,t_1])}+E^1_{interp}+E^1_{integr},
\end{aligned} \tag{10.21}$$

where

$$\begin{aligned}
E^1_{interp} := \sum_{s=0,\dots,S_k-1}\Bigg(&\frac{\varphi(\widehat{u}_{j,\,s+1},z_{s+1})-\varphi(\widehat{u}_{j,\,s},z_s)}{8}\Delta_s\\
&+\frac{2}{3}\Delta_s\left[\mathrm{L}_{1,s}|\widehat{u}_{j,s+1}-\widehat{u}_{j,s}|+\mathrm{L}_{2,s}\Delta_s\right]\Bigg)
\end{aligned} \tag{10.22}$$

and

$$E^1_{integr} := \sum_{s=0,\dots,S_k-1}\left(\frac{\mathrm{L}_s}{2}\Delta_s^2-\frac{1}{2\mathrm{L}_s}\left[\varphi(\widehat{u}_{j,\,s+1},z_{s+1})-\varphi(\widehat{u}_{j,\,s},z_s)\right]^2\right), \tag{10.23}$$

where for $j=0$ the function $\widehat{u}_j$ is taken as a piecewise affine interpolation of u_0, and for $j\ge 1$ it is taken from the previous iteration step.

The quantity $\mathrm{M}_j^{\oplus,1}$ is fully computable, and it shows the overall error associated with the step number j on the first interval.

Remark 10.2 Estimate of the overall error related to the interval $[t_0,t_K]$ includes all errors computed on the intervals. In other words the error associated with $[t_0,t_{k-1}]$ is appended to the error on $[t_{k-1},t_k]$ (which formally follows from the fact that the initial condition on $[t_{k-1},t_k]$ includes errors on the previous intervals).

Thus, we have shown that fully guaranteed and computable bounds can indeed be derived for the problem (10.1) with the Lipschitz function φ, i.e. for every finite time interval $[t_0, t_K]$ and for every a priori required accuracy ε an approximate solution of the problem can be found by the APL method discussed above.

10.5 The APL Algorithm and Numerical Examples

Let ε be a required accuracy of an approximate solution. Then, practical computation can be performed by Algorithm 10.1.

Algorithm 10.1 The algorithm of the APL method

Input: ε {required accuracy on the interval} , u_0 {input initial boundary condition}

$\mathscr{F}_K = \bigcup_{k=0}^{K-1} [t_k, t_{k+1}]$ { constructed by *Mesh Generation Procedure*}

$\varepsilon^k = \frac{\varepsilon}{K}$ {obtain accuracy of the approximate solution on interval $[t_k, t_{k+1}]$}

$\Omega_{S_k} = \bigcup_{s=0}^{S_k-1} [z_s, z_{s+1}]$ {initial mesh for each subinterval}

for $k = 1$ to K **do**

  $j = 0$

  do

    if $k = 1$

      $a = u_0$

    else

      $a = v^{k-1}(t_{k-1})$

    endif

    $v_j^k = Integration\ Procedure(\varphi, v_{j-1}^k, S_k) + a$

    calculate E_{interp}^k and E_{integr}^k by using (10.22) and (10.23)

    $M_j^{\oplus,k} = \|v_j^k - v_{j-1}^k\|_{C([t_{k-1},t_k])} + E_{interp}^k + E_{integr}^k$

    $e_j^{\oplus} = \frac{q}{1-q} M_j^{\oplus,k}$

    if $E_{interp}^k + E_{integr}^k > \varepsilon^k$

      $S_k = 2\, S_k$ {refine the mesh Ω_{S_k}}

    endif

    $j = j + 1$

  while $e_j^{\oplus} > \varepsilon^k$

  $v^k = v_j^k$ {the approximate solution on the interval $[t_{k-1}, t_k]$}

  $e^{\oplus,k} = e_j^{\oplus}$ {error bound achieved for the interval $[t_{k-1}, t_k]$}

end for

Output: $\{v^k\}_{k=1}^K$ {the approximate solution}

$\{e^{\oplus,k}\}_{k=1}^K$ {error bounds estimates on sub intervals}

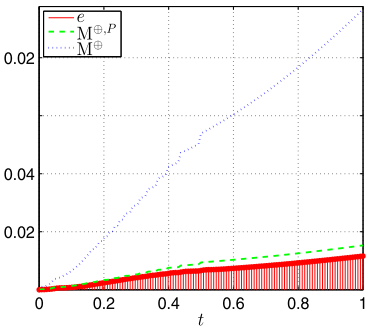


Fig. 10.1 The error and error majorants

In general, the algorithm should start with the generation of a suitable mesh (i.e., select time intervals). Here, we do not discuss this question in detail, but only note that the *Mesh Guaranteed Procedure* must adapt the mesh to the nature of $\varphi(u(t), t)$, which requires information about U (see (10.3)). In practise, such information can be obtained by solving the problem (10.1) numerically with the help of some heuristic (e.g., Runge-Kutta) method on a coarse mesh.

The APL algorithm is a cycle over all the intervals of the mesh $\mathscr{T}_K = \bigcup_{k=0}^{K-1} [t_k, t_{k+1}]$. On each subinterval, the algorithm is realized as a subcycle (whose index is j). In the subcycle, we apply the PL method and try to find an approximation that meets the accuracy requirements imposed (i.e., the accuracy must be higher than ε^k). Initial data are taken from the previous step (for the first step, the initial condition is defined by u_0).

After computing an approximation on $[t_k, t_{k+1}]$ we use our majorant and find a guaranteed upper bound (which includes the interpolation and integration errors). Iterations are continued unless the required accuracy ε^k has been achieved. After that we save the results and proceed to the next interval.

Note that in Algorithm 10.1, we do not discuss in detail the process of integration on an interval, which is performed on a local mesh with a certain amount of subintervals (whose size is Δ_s). In principle, it may happen that the desired level of accuracy, ε^k, is not achieved with the Δ_s selected. This fact will be easily detected because interpolation and integration errors will dominate and do not allow the overall error to decrease below ε^k. In this case, Δ_s must be reduced, and computations on the corresponding interval must be repeated.

Example 10.1 Consider the problem

$$\frac{du}{dt} = 4ut\sin(8t), \quad t \in [0, 3/2],$$
$$u(0) = u_0 = 1$$

with the exact solution

$$u = e^{\frac{1}{16}\sin(8t) - \frac{1}{2}t\cos(8t)}.$$

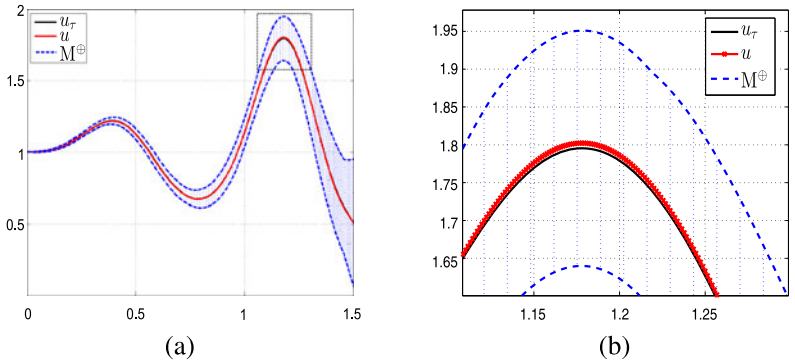


Fig. 10.2 (**a**) The exact and approximate solutions with guaranteed bounds of the deviation computed by the Ostrowski estimate. (**b**) A zoomed interval of the exact and approximate solutions with bounds of the deviation computed by the majorant

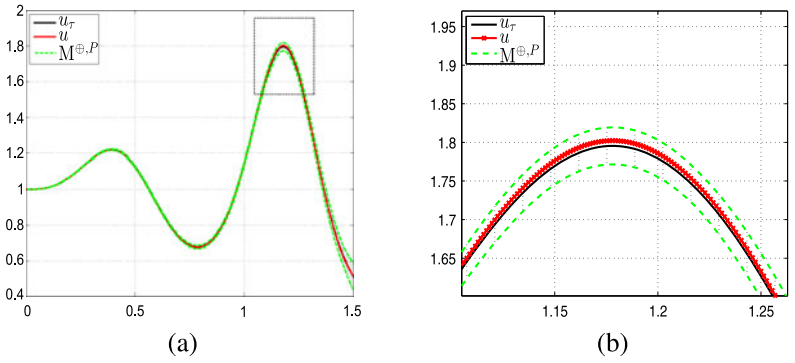


Fig. 10.3 (**a**) The exact and approximate solutions with guaranteed bounds of the deviation computed by the advanced form of the estimate. (**b**) A zoomed interval of the exact and approximate solutions with bounds of the deviation computed by the majorant

In Fig. 10.1, we depict the error (bold dots), error bounds computed by the Ostrowski estimates (dotted line) and by the advanced form of the estimate (dashed line). In order to make the results more transparent, we depict the approximate solution together with the zone which contains the exact solution (see Figs. 10.2(a) and 10.3(a)). The form of this (shaded) zone is determined by the a posteriori estimates.

Thus, the APL method computes two-sided guaranteed bounds containing the exact solution. It may happen that the desired level of accuracy has been exceeded at some moment $t' < t_K$ and further Picard-Lindelöf iterations are unable to reduce the error. This situation may arise if the amount of internal points used for numerical integration on each interval is too small. In this case, we must enlarge the number of internal nodes (which will reduce integration and interpolation errors) and repeat the computations. Numerical results illustrated in Figs. 10.2(b) and

Table 10.1 Components of the general estimate

Estimate of $\|e_j\|$	Estimate of $\|\bar{e_j}\|$	Estimate of $\|\hat{e_j}\|$
2.2658e-002	8.6160e-008	9.5725e-008
4.6095e-002	1.8847e-007	5.8148e-007
5.4949e-002	2.5299e-007	5.9301e-007
7.4818e-002	2.5768e-007	2.3618e-006
9.5993e-002	3.0190e-007	2.3699e-006
1.0302e-001	3.4216e-007	2.3807e-006
1.5427e-001	4.8963e-007	2.4320e-006
1.5647e-001	6.1877e-007	2.4999e-006
2.3495e-001	9.4891e-007	2.6183e-006
2.7145e-001	9.8935e-007	2.6328e-006
3.0533e-001	9.9923e-007	2.6373e-006
3.2838e-001	1.0158e-006	2.6404e-006
4.4629e-001	1.0182e-006	2.6517e-006

10.3(b) show that the advanced majorant provides much sharper bounds of the deviation.

Values of the components of the estimate (the first term, the *estimate of* $\|\bar{e}\|$ and the *estimate of* $\|\widehat{e}\|$ from (10.21)) are presented in Table 10.1. We see that in this example the values of S_k were selected properly, so that interpolation and integration error estimates are insignificant with respect to the first term.

Example 10.2 The APL method works with stiff problems as well. Consider the classical stiff equation

$$\frac{du}{dt} = 50\cos(t) - 50u, \quad t = [0, 1],$$
$$u(0) = u_0 = 1$$

with the exact solution

$$u = \frac{1}{2501}e^{-50t} + \frac{2500}{2501}\cos(t) + \frac{50}{2501}\sin(t).$$

Analogously to the previous example, in Fig. 10.4(a) the general error (lines with dots on the top) estimated by the Ostrowski estimate (dotted line) and the advanced form of the estimate (dashed line) are illustrated. Another way to depict obtained results is shown in Fig. 10.4(b).

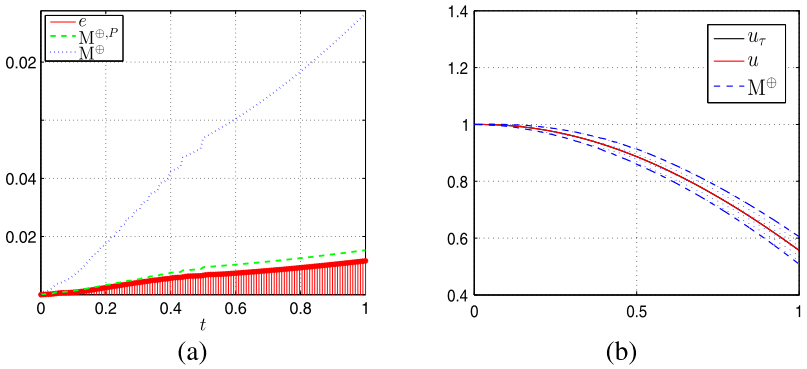


Fig. 10.4 (**a**) The error and error majorants. (**b**) The exact and approximate solutions with the guaranteed deviation bound

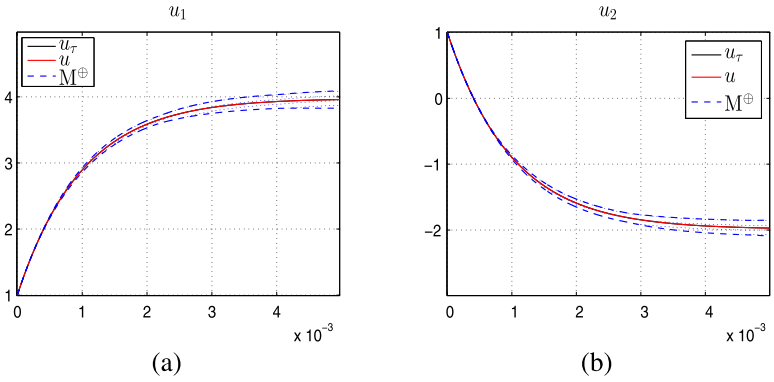


Fig. 10.5 The exact and approximate solutions of the system and the guaranteed error bounds computed by the Ostrowski method

Example 10.3 The APL method can also be applied to stiff systems of ODEs. As an example, we consider the system

$$\begin{cases} \dfrac{du_1}{dt} = 998u_1 + 1998u_2, \\ \dfrac{du_2}{dt} = -999u_1 - 1999u_2, \\ u_1(t_0) = 1, \qquad u_2(t_0) = 1, \\ t \in [0, 5 \times 10^{-3}] \end{cases}$$

with the exact solutions $u_1 = 4e^{-t} - 3e^{-1000t}$ and $u_2 = -2e^{-t} + 3e^{-1000t}$. In Figs. 10.5(a), 10.5(b), 10.6(a), and 10.6(b), we present the same type of information (behavior of the solution and guaranteed bounds) as in the previous examples.

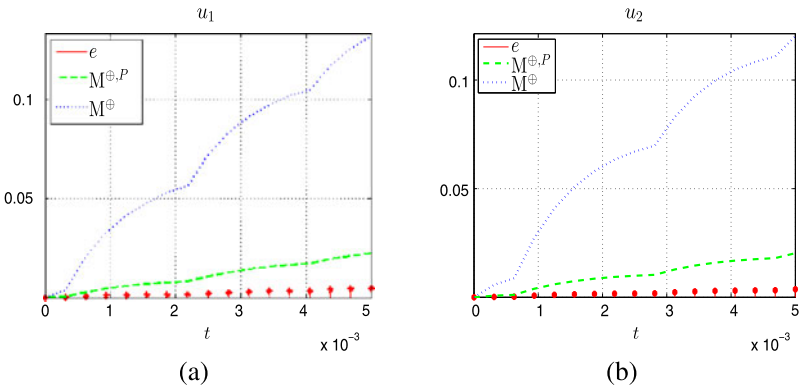


Fig. 10.6 The error and error majorants for the solutions u_1, u_2 of the system

We note that for stiff equations getting an approximate solution with the guaranteed and sharp error bounds requires much larger expenditures than in relatively simple Examples 10.1 and 10.2. This result is not surprising because (as it is quite natural to expect) for such type of problems fully reliable computations will be much more expensive.

References

1. Coddington EA, Levinson N (1972) Theory of ordinary differential equations. McGraw-Hill, New York
2. Eirola T, Krasnosel'skii AM, Krasnosel'skii MA, Kuznetsov NA, Nevanlinna O (1995) Incomplete corrections in nonlinear problems. Nonlinear Anal 25(7):717–728
3. Lindelöf E (1894) Sur l'application de la méthode des approximations successives aux équations différentielles ordinaires du premier ordre. C R Hebd Séances Acad Sci 114:454–457
4. Neittaanmäki P, Repin S (2004) Reliable methods for computer simulation. Error control and a posteriori estimates. Elsevier, Amsterdam
5. Nevanlinna O (1989) Remarks on Picard-Lindelöf iteration. Part I. BIT Numer Math 29(2):328–346
6. Nevanlinna O (1989) Remarks on Picard-Lindelöf iteration. Part II. BIT Numer Math 29(3):535–562
7. Ostrowski A (1972) Les estimations des erreurs a posteriori dans les procédés itératifs. C R Hebd Séances Acad Sci Séries A et B 275:A275–A278
8. Repin S (2008) A posteriori estimates for partial differential equations. Walter de Gruyter, Berlin

Part III
Analysis of Noised and Uncertain Data

Chapter 11
Hermitian Interpolation Subject to Uncertainties

Jean-Antoine Désidéri, Manuel Bompard, and Jacques Peter

Abstract This contribution is a sequel of the report (Bompard et al. in http://hal.inria.fr/inria-00526558/en/, 2010). In PDE-constrained global optimization (e.g., Jones (in J. Global Optim. 21(4):345–383, 2001)), iterative algorithms are commonly efficiently accelerated by techniques relying on approximate evaluations of the functional to be minimized by an economical but lower-fidelity model ("meta-model"), in a so-called "Design of Experiment" (DoE) (Sacks et al. in Stat. Sci. 4(4):409–435, 1989). Various types of meta-models exist (interpolation polynomials, neural networks, Kriging models, etc.). Such meta-models are constructed by pre-calculation of a database of functional values by the costly high-fidelity model. In adjoint-based numerical methods, derivatives of the functional are also available at the same cost, although usually with poorer accuracy. Thus, a question arises: should the derivative information, available but known to be less accurate, be used to construct the meta-model or be ignored? As the first step to investigate this issue, we consider the case of the Hermitian interpolation of a function of a single variable, when the function values are known exactly, and the derivatives only approximately, assuming a uniform upper bound ϵ on this approximation is known. The classical notion of best approximation is revisited in this context, and a criterion is introduced to define the best set of interpolation points. This set is identified by either analytical or numerical means. If $n+1$ is the number of interpolation points, it is advantageous to account for the derivative information when $\epsilon \leq \epsilon_0$, where ϵ_0 decreases with n, and this is in favor of piecewise, low-degree Hermitian interpolants. In all our numerical tests, we have found that the distribution of Chebyshev points

J.-A. Désidéri (✉)
INRIA, Centre de Sophia Antipolis – Méditerranée, BP 93, 2004 Route des Lucioles, 06902 Sophia Antipolis cedex, France
e-mail: Jean-Antoine.Desideri@inria.fr

M. Bompard · J. Peter
ONERA/DSNA, BP 72, 29, Avenue de La Division Leclerc, 91322 Châtillon cedex, France

M. Bompard
e-mail: Manuel.Bompard@onera.fr

J. Peter
e-mail: Jacques.Peter@onera.fr

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_11,

is always close to optimal, and provides bounded approximants with close-to-least sensitivity to the uncertainties.

11.1 Introduction: The Classical Notion of Best Approximation

In this section, we review certain classical notions on polynomial interpolation, in particular the concept of "best approximation" or "Chebyshev economization". The literature contains numerous elementary and advanced texts on this fundamental issue, and we refer to [2, 4, 5].

Let n be an integer and $x_0, x_1, \ldots, x_n$ be $n+1$ distinct points of the normalized interval $[-1, 1]$. Let $\pi(x)$ be the following polynomial of degree $n+1$:

$$\pi(x) = \prod_{i=0}^{n} (x - x_i)$$

and consider the following $n+1$ polynomials of degree n:

$$\mathsf{L}_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \quad (i = 0, 1, \ldots, n).$$

Clearly

$$\forall i, j \in \{0, 1, \ldots, n\} \ : \ \mathsf{L}_i(x_j) = \delta_{i,j},$$

where δ stands for Krönecker's symbol. Application of L'Hospital's rule yields the following compact formula:

$$\mathsf{L}_i(x) = \frac{\pi(x)}{\pi'(x_i)(x - x_i)}. \tag{11.1}$$

Let $f : [-1, 1] \to \mathbb{R}$ be a smooth function of the real variable x. The polynomial

$$P_n(x) = \sum_{i=0}^{n} f(x_i) \mathsf{L}_i(x)$$

is of degree at most equal to n, and it clearly satisfies the following interpolation conditions:

$$\forall i \in \{0, 1, \ldots, n\} \ : \ P_n(x_i) = f(x_i).$$

One such interpolant is unique among all polynomials of degree $\leq n$. Thus, $P_n(x)$ is the Lagrange interpolation polynomial of f at the points $\{x_i\}_{0 \leq i \leq n}$.

It is well known that if $f \in \mathsf{C}^{n+1}([-1, 1])$, for any given $x \in [a, b]$, the interpolation error is given by

$$e_n(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \pi(x)$$

for some $\xi \in [-1, 1]$.

Proof Let $x \in [-1, 1]$ be given. If $x = x_i$ for some i, the result is trivially satisfied. Otherwise, $\pi(x) \neq 0$; then let

$$\lambda = \frac{e_n(x)}{\pi(x)}$$

so that

$$f(x) = P_n(x) + \lambda \pi(x)$$

and define the function

$$\phi(t) = f(t) - P_n(t) - \lambda \pi(t) \quad t \in [-1, 1].$$

The function $\phi(t)$ is of class $\mathsf{C}^{n+1}([-1, 1])$ and it admits a nonempty set of roots in the interval $[-1, 1]$ that includes $X = \{x_0, x_1, \ldots, x_n, x\}$. The $n + 2$ elements of X are distinct and can be arranged as the elements of a strictly increasing sequence $\{x_i^0\}_{0 \leq i \leq n+1}$ whose precise definition depends on the position of x w.r.t. the interpolation points $\{x_i\}_{0 \leq i \leq n}$. By application of Rolle's theorem to $\phi(t) = \phi^{(0)}(t)$ over the subinterval $[x_i^0, x_{i+1}^0]$, $i = 0, 1, \ldots, n$, it follows that $\phi'(t)$ admits a root x_i^1 in the open interval $]x_i^0, x_{i+1}^0[$, and this, for each i. In this way we identify a strictly-increasing sequence of $n + 1$ roots of $\phi'(t)$, $\{x_i^1\}_{0 \leq i \leq n}$. Then Rolle's theorem can be applied in a similar way, this time to $\phi'(t)$, and so on to the successive derivatives of $\phi(t)$. We conclude that in general $\phi^{(k)}(t)$ admits at least $n + 2 - k$ distinct roots in $[-1, 1]$, $\{x_i^k\}_{0 \leq i \leq n+1-k}$, $0 \leq k \leq n + 1$. In particular, for $k = n + 1$, $\phi^{(n+1)}(t)$ admits at least one root, x_0^{n+1}, hereafter denoted ξ for simplicity. But since $P_n^{(n+1)}(\xi) = 0$ and $\pi^{(n+1)}(\xi) = (n + 1)!$, one gets

$$\lambda = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

and the conclusion follows. □

Hence, if

$$K = \frac{1}{(n+1)!} \max_{x \in [-1,1]} \left| f^{(n+1)}(x) \right|$$

we have

$$\forall x \in [-1, 1] \; : \; \left| e_n(x) \right| \leq K \left| \pi(x) \right|.$$

Therefore, a natural way to optimize the choice of interpolation points *a priori*, that is, independently of f, is to solve the following classical min-max problem:

$$\min_{\substack{\{x_i\}_{0 \leq i \leq n} \\ x_i \in [-1,1], \; \forall i}} \max_{x \in [-1,1]} \left| \pi(x) \right|. \tag{11.2}$$

In view of this, the problem is to find among all polynomials $\pi(x)$ whose highest-degree term is precisely x^{n+1}, and that admit $n+1$ distinct roots in the interval $[-1, 1]$, an element, unique or not, that minimizes the sup-norm over $[-1, 1]$.

The solution of this problem is given by the $n+1$ roots of the Chebyshev polynomial of degree $n+1$. Before recalling the proof of this, let us establish some useful auxiliary results. Let k be an arbitrary integer and $T_k(x)$ denote the Chebyshev polynomial of degree k. Recall that for $x \in [-1, 1]$

$$T_k(x) = \cos\bigl(k \cos^{-1} x\bigr), \quad k \in \mathbb{N},$$

so that, for $k \geq 1$ and $x \in [-1, 1]$,

$$T_{k+1}(x) + T_{k-1}(x) = \cos(\overline{k+1}\theta) + \cos(\overline{k-1}\theta) = 2\cos(k\theta)\cos\theta = 2xT_k(x),$$

where one has let

$$x = \cos\theta, \quad 0 \leq \theta \leq \pi.$$

Thus, if the leading term in $T_k(x)$ is say $a_k x^k$, the following recursion applies:

$$a_{k+1} = 2a_k, \quad k \geq 1,$$

and, since $a_0 = a_1 = 1$, it follows that

$$a_k = 2^{k-1}, \quad k \geq 1.$$

Therefore, an admissible candidate solution for the min-max problem, (11.2), is the polynomial

$$\pi^\star(x) = \frac{1}{2^n} T_{n+1}(x).$$

It remains to establish that $\pi^\star(x)$ is the best choice among all admissible polynomials, and its roots the best possible interpolation points. To arrive at this, we claim the following lemma:

Lemma 11.1 *For all admissible polynomial $q(x)$ one has*

$$\|\pi^\star\| \leq \|q\|,$$

where $\|\ \|$ is the sup-norm over $[-1, 1]$.

Proof Assume otherwise that an admissible polynomial $q(x)$ of a strictly smaller sup-norm over $[-1, 1]$ exists:

$$\|q\| < \|\pi^\star\|.$$

Let $r(x) = \pi^\star(x) - q(x)$. Since the admissible polynomials $\pi^\star(x)$ and $q(x)$ have the same leading term, x^{n+1}, the polynomial $r(x)$ is of degree at most n. Let us

examine the sign of this polynomial at the $n+2$ points

$$\eta_i = \cos\frac{i\pi}{n+1}, \quad i = 0, 1, \ldots, n+1,$$

at which $\pi^\star(x)$ as well as $T_{n+1}(x)$ achieve a local extremum. At such a point,

$$\left|\pi^\star(\eta_i)\right| = \frac{1}{2^n} = \left\|\pi^\star\right\| > \|q\| = \max_{x\in[-1,1]}\left|q(x)\right| \geq \left|q(\eta_i)\right|$$

and $r(\eta_i)$ is nonzero and has the sign of the strictly dominant term $\pi^\star(\eta_i) = \frac{1}{2^n}T_{n+1}(\eta_i) = \frac{(-1)^i}{2^n}$. Therefore, $r(x)$ admits at least $n+1$ sign alternations and as many distinct roots. But this is in contradiction with the degree of this polynomial. The contradiction is removed by rejecting the assumption made on $\|q\|$. □

Consequently, in (11.2), the min-max is achieved by the roots of $T_{n+1}(x)$:

$$x_i^\star = \cos\frac{(2i+1)\pi}{2(n+1)}, \quad i = 0, 1, \ldots, n, \tag{11.3}$$

and the value of the min-max is $\frac{1}{2^n}$.

11.2 Best Hermitian Approximation

Now assume that the points $\{x_i\}_{0\leq i\leq n}$ are used as a support to interpolate the function values $\{y_i = f(x_i)\}_{0\leq i\leq n}$, but also the derivatives $\{y_i' = f'(x_i)\}_{0\leq i\leq n}$, that is a set of $2(n+1)$ data. Thus, we anticipate that the polynomial of least degree complying with these interpolation conditions, say $H_{2n+1}(x)$, is of degree at most equal to $2n+1$. One such polynomial is necessarily of the form

$$H_{2n+1}(x) = P_n(x) + \pi(x)\cdot Q(x), \tag{11.4}$$

where the quotient $Q(x)$ should be adjusted to comply with the interpolation conditions on the derivatives. These conditions are

$$y_i' = H_{2n+1}'(x_i) = P_n'(x_i) + \pi_i'\cdot Q(x_i), \quad i = 0, 1, \ldots, n, \tag{11.5}$$

where

$$\pi_i' = \pi'(x_i) = \prod_{\substack{j=0\\ j\neq i}} (x_i - x_j) \neq 0, \tag{11.6}$$

and since $\pi(x_i) = 0$. Thus $Q(x)$ is solely constrained by the following $n+1$ interpolation conditions:

$$Q_i = Q(x_i) = \frac{y_i' - P_n'(x_i)}{\pi_i'}, \quad i = 0, 1, \ldots, n. \tag{11.7}$$

Therefore, the solution of least degree is obtained when $Q(x)$ is the Lagrange interpolation polynomial associated with the above function values:

$$Q(x) = \sum_{i=0}^{n} Q_i \mathsf{L}_i(x).$$

The corresponding solution is thus unique, and we will refer to it as the global Hermitian interpolant.

The interpolation error associated with the above global Hermitian interpolant $H_{2n+1}(x)$ is given by the following result valid when $f \in \mathsf{C}^{2n+2}([-1,1])$:

$$\forall x \in [-1,1],\ \exists \eta \in [-1,1]\ :\ f(x) = H_{2n+1}(x) + \frac{f^{(2n+2)}(\eta)}{(2n+2)!}\,\pi^2(x). \qquad (11.8)$$

Proof Let $x \in [-1,1]$ be given. If $x = x_i$ for some i, the result is trivially satisfied. Otherwise, $\pi(x) \neq 0$; then let

$$\mu = \frac{f(x) - H_{2n+1}(x)}{\pi^2(x)}$$

so that

$$f(x) = H_{2n+1}(x) + \mu\pi^2(x).$$

Let

$$\psi(t) = f(t) - H_{2n+1}(t) - \mu\pi^2(t), \quad t \in [-1,1].$$

The function $\psi(t)$ is of class $\mathsf{C}^{2n+2}([-1,1])$, and similarly to the former function $\phi(t)$, it admits a nonempty set of roots in the interval $[-1,1]$ that includes $X = \{x_0, x_1, \ldots, x_n, x\} = \{x_i^0\}_{0\le i\le n+1}$. Hence, Rolle's theorem implies that in the open interval $]x_i, x_{i+1}[$, $0 \le i \le n$, a root x_i' of $\psi'(t)$ exists. But the interpolation points, at which the derivative also is fitted, are themselves $n+1$ other distinct roots of $\psi'(t)$. Thus we get a total of at least $2n+2$ roots for $\psi'(t)$, and by induction, $2n+1$ for $\psi''(t)$, and so on, and finally one, say η, for $\psi^{(2n+2)}(t)$. Now, since $H_{2n+1}^{(2n+2)}(\eta) = 0$ because the interpolant is of degree $2n+1$ at most, and since $(\pi^2(t))^{(2n+2)}(t) = (2n+2)!$, one gets

$$0 = f^{(2n+2)}(\eta) - 0 - \mu\,(2n+2)!$$

which yields the final result. □

As a consequence of (11.8), the formulation of the best approximation problem for the global Hermitian interpolant is as follows:

$$\min_{\substack{\{x_i\}_{0\le i\le n} \\ x_i\in[-1,1],\ \forall i}} \max_{x\in[-1,1]} \pi^2(x). \qquad (11.9)$$

But, if we define the following functions of $\mathbb{R}^{n+1} \to \mathbb{R}$:

$$\begin{cases} P(x_0, x_1, \ldots, x_n) = \max\limits_{x \in [-1,1]} \pi^2(x), \\ p(x_0, x_1, \ldots, x_n) = \max\limits_{x \in [-1,1]} \left|\pi(x)\right|, \end{cases}$$

it is obvious that

$$\forall x_0, x_1, \ldots, x_n \ : \ P(x_0, x_1, \ldots, x_n) = p^2(x_0, x_1, \ldots, x_n).$$

Hence the functions P and p achieve their minimums for the same sequence of interpolation points, and

$$\min_{\substack{\{x_i\}_{0 \le i \le n} \\ x_i \in [-1,1],\ \forall i}} P(x_0, x_1, \ldots, x_n) = \Big(\min_{\substack{\{x_i\}_{0 \le i \le n} \\ x_i \in [-1,1],\ \forall i}} p(x_0, x_1, \ldots, x_n) \Big)^2 = \frac{1}{4^n}.$$

Therefore the best Hermitian interpolation is achieved for the same set of interpolation points as the best Lagrangian interpolation, that is, the roots, $\{x_i^\star\}_{0 \le i \le n}$ of (11.3), of the Chebyshev polynomial $T_{n+1}(x)$.

11.3 Best Inexact Hermitian Approximation

In PDE-constrained global optimization [3], it is often useful to model the functional criterion to be optimized by a function $f(x)$ of the design variable $x \in \mathbb{R}^n$, whose values are computed through the discrete numerical integration of a PDE, and the derivative $f'(x)$, or gradient vector, by means of an adjoint equation. A database of function values and derivatives is compiled by Design of Experiment, and the surrogate model, or meta-model is constructed from it. This meta-model is then used in some way in the numerical optimization algorithm with the objective of improving computational efficiency (see, for example, [3]). The success of such a strategy depends on the accuracy of the meta-model $f(x)$ to represent the dependency on x of the actual functional criterion. If all the data were exact, and properly used, the accuracy would undoubtedly improve by the addition of the derivative information. However, in practice, since the PDE is solved discretely, the derivatives are almost inevitably computed with inferior accuracy. Therefore it is not clear that accounting for the derivatives is definitely advantageous if the corresponding accuracy of the data is poor. Should special precautions be taken to guarantee it?

In order to initiate a preliminary analysis of this problem, we examine the simple one-dimensional situation of a function $f(x)$, when x is scalar ($x \in \mathbb{R}$), and consider the case of a Hermitian interpolation meta-model based on inexact information. Specifically, we assume that the function values $\{y_i\}_{0 \le i \le n}$ are known, whereas only approximations $\{\bar{y}_i'\}_{0 \le i \le n}$ of the derivatives $\{y_i'\}_{0 \le i \le n}$ are available, and we let:

$$\delta y_i' = \bar{y}_i' - y_i' := \epsilon_i, \quad i = 0, 1, \ldots, n.$$

Hence the computed interpolant is $\overline{H}_{2n+1}(x)$ instead of $H_{2n+1}(x)$, and in view of the definitions (11.4)–(11.7), we have

$$\begin{cases} \delta H_{2n+1}(x) = \overline{H}_{2n+1}(x) - H_{2n+1}(x) = \pi(x)\,\delta Q(x), \\ \delta Q(x) = \displaystyle\sum_{i=0}^{n} \delta Q_i \mathsf{L}_i(x), \\ \delta Q_i = \dfrac{\delta y_i'}{\pi_i'} = \dfrac{\epsilon_i}{\pi_i'}. \end{cases} \tag{11.10}$$

Now, suppose an upper bound ϵ on the errors ϵ_i's is known:

$$|\epsilon_i| \le \epsilon, \quad 0 \le i \le n. \tag{11.11}$$

The following questions arise:

1. What is the corresponding upper bound on $\max_{x\in[-1,1]} |\delta H_{2n+1}(x)|$?
2. Can we choose the sequence of interpolation points $\{x_i\}_{0\le i\le n}$ to minimize this upper bound?
3. Is the known sequence of the Chebyshev points a good approximation of the optimum sequence for this new problem?

This article attempts to bring some answers to these questions. Presently, we try to identify how the interpolation points should be defined to minimize or reduce the effect on the meta-model accuracy of uncertainties on the derivatives only. It follows from (11.10) that

$$\delta H_{2n+1}(x) = \pi(x) \sum_{i=0}^{n} \frac{\epsilon_i}{\pi_i'} \mathsf{L}_i(x)$$

which by virtue of (11.1) simplifies as follows:

$$\delta H_{2n+1}(x) = \pi^2(x) \sum_{i=0}^{n} \frac{\epsilon_i}{\pi_i'^2\,(x - x_i)}.$$

Thus if (11.11) holds, we have

$$\left|\delta H_{2n+1}(x)\right| \le \epsilon\, \Delta(x)$$

where

$$\Delta(x) = \pi^2(x) \sum_{i=0}^{n} \frac{1}{\pi_i'^2\,|x - x_i|}.$$

These considerations have led us to analyze the min-max problem applied to the new criterion $\Delta(x)$ in place of $\pi^2(x)$. In summary, the solution of the min-max problem associated with the criterion $\Delta(x)$ minimizes the effect of uncertainties in

the derivatives on the identification of the global Hermitian interpolant. In the subsequent sections, this solution is identified formally, or numerically, and compared with the Chebyshev distribution of points, which is optimal w.r.t. the interpolation error. Lastly, the corresponding interpolants are compared by numerical experiment.

11.4 Formal or Numerical Treatment of the Min-Max-Δ Problem

We wish to compare three particular relevant distributions of interpolation points in terms of performance w.r.t. the criterion $\Delta(x)$. These three distributions are symmetrical w.r.t. 0, and recall that the total number of interpolation points is $n+1$. Thus, we let

$$n+1 = 2m + \alpha$$

and when n is odd ($\alpha = 0$; $n = 2m-1 \geq 1$),

$$\{x_i\}_{0 \leq i \leq n} = \{\pm\xi_1, \pm\xi_2, \ldots, \pm\xi_m\},$$

where

$$0 < \xi_1 < \xi_2 < \cdots < \xi_m$$

and $m \geq 1$. Otherwise, when n is even ($\alpha = 1$; $n = 2m \geq 0$), we adjoin to the list $\xi_0 = 0$ (once). We consider specifically:

1. The uniform distribution:

   $n = 2m$: $\xi_0^u = 0$ associated with the interpolation point $x_0 = \xi_0 = 0$, and $\xi_k^u = \frac{k}{m}$, $1 \leq k \leq m$, associated with 2 interpolation points $\pm\xi_k^u$.
   $n = 2m-1$: $\xi_k^u = \frac{2k-1}{n}$, $1 \leq k \leq m$.

2. The Chebyshev distribution:

$$\xi_k^\star = x_{m-k}^\star = \cos\left(\frac{2(m-k)+1}{n+1}\,\frac{\pi}{2}\right), \quad 1 \leq k \leq m.$$

3. The optimal distribution:

$$\bar{\xi} = \arg\min_{\xi} \max_{x \in [0,1]} \Delta(x;\xi),$$

   where $\xi = (\xi_1, \xi_2, \ldots, \xi_m)$ denotes the vector of adjustable parameters defining, along with $\xi_0 = 0$ if n is even, the distribution of interpolation points, and the dependence of the criterion Δ on ξ is here indicated explicitly for clarity. (Note that due to symmetry, the interval for x has been reduced to [0, 1] without incidence on the result.)

To these three distributions are associated the corresponding values of the maximum of $\Delta(x;\xi)$ over $x \in [0,1]$; these maximums are denoted Δ^u, $\Delta^\star$ and $\bar{\Delta}$, respectively.

As a result of these definitions, the polynomial $\pi(x)$ is expressed as follows:

$$\pi(x) = x^\alpha \prod_{k=1}^{m} \left(x^2 - \xi_k^2\right), \quad n+1 = 2m+\alpha;\ \alpha = 0 \text{ or } 1,$$

and for $x > 0$, the criterion $\Delta(x)$ becomes

$$\Delta(x) = \pi^2(x) \sum_{i=0}^{n} \frac{1}{\pi_i'^2\,|x - x_i|} = \pi^2(x) \left[\frac{\alpha}{\pi_0'^2} \frac{1}{x} + \sum_{k=1}^{m} \frac{1}{\pi_k'^2} \left(\frac{1}{x+\xi_k} + \frac{1}{|x-\xi_k|} \right) \right].$$

Then, given x, let j be the index for which

$$\xi_{j-1} \le x < \xi_j$$

so that

$$x - \xi_k \ge 0 \Longleftrightarrow k \le j-1.$$

As a result,

$$\Delta(x) = \pi^2(x) \left[\frac{\alpha}{\pi_0'^2} \frac{1}{x} + \sum_{k=1}^{j-1} \frac{1}{\pi_k'^2} \frac{2x}{x^2 - \xi_k^2} + \sum_{k=j}^{m} \frac{1}{\pi_k'^2} \frac{2\xi_k}{\xi_k^2 - x^2} \right]. \tag{11.12}$$

Calculation of the derivatives π_k' First, for $\alpha = 0$, $\pi(x)$ is an even polynomial, and $\pi_0' = 0$. Otherwise, for $\alpha = 1$,

$$\pi_0' = \lim_{x\to 0} \frac{\pi(x)}{x} = \prod_{k=1}^{m} \left(-\xi_k^2\right), \quad \alpha = 1;\ n = 2m.$$

Regardless α, for $k \ge 1$

$$\pi(x) = x^\alpha (x - \xi_k)(x + \xi_k) \prod_{\substack{i=1 \\ i\neq k}} \left(x^2 - \xi_i^2\right)$$

so that:

$$\pi_k' = \lim_{x\to \xi_k} \frac{\pi(x)}{x - \xi_k} = 2\xi_k^{\alpha+1} \prod_{\substack{i=1 \\ i\neq k}} \left(\xi_k^2 - \xi_i^2\right).$$

11.4.1 Expression of δ_0 Applicable Whenever $n+1$ is Even ($\alpha = 0$; $n+1 = 2m$)

Suppose that $0 < x < \xi_1$. Then $j = 1$ and (11.12) reduces to

$$\Delta(x) = \pi^2(x) \sum_{k=1}^{m} \frac{1}{\pi_k'^2} \frac{2\xi_k}{\xi_k^2 - x^2}.$$

But,

$$\pi^2(x) = \prod_{i=1}^{m} \left(x^2 - \xi_i^2\right)^2$$

so that

$$\Delta(x) = \sum_{k=1}^{m} \frac{2\xi_k}{\pi_k'^2} \times \prod_{\substack{i=1 \\ i \neq k}}^{m} \left(\xi_i^2 - x^2\right)^2 \times \left(\xi_k^2 - x^2\right).$$

All the terms in this sum are composed of three factors: a positive constant and two positive factors that are monotone decreasing as x varies from 0 to ξ_1. Hence $\Delta(x)$ is monotone decreasing and

$$\begin{aligned} \delta_0 &= \max_{x \in [0, \xi_1]} \Delta(x) = \Delta(0) \\ &= \sum_{k=1}^{m} \frac{2\xi_k^3}{\pi_k'^2} \prod_{\substack{i=1 \\ i \neq k}}^{m} \xi_i^4 = 2\left(\prod_{i=1}^{m} \xi_i^4\right) \sum_{k=1}^{m} \frac{1}{\xi_k \pi_k'^2}, \quad \alpha = 0;\ n+1 = 2m. \end{aligned} \tag{11.13}$$

11.4.2 Expression of δ_1 Applicable Whenever $\xi_m < 1$

Suppose that $\xi_m < x < 1$. Then $j = m+1$ and (11.12) reduces to

$$\begin{aligned} \Delta(x) &= \pi^2(x) \left[\frac{\alpha}{\pi_0'^2} \frac{1}{x} + \sum_{k=1}^{m} \frac{1}{\pi_k'^2} \frac{2x}{x^2 - \xi_k^2} \right] \\ &= \frac{\alpha}{\pi_0'^2} x \prod_{i=1}^{m} \left(x^2 - \xi_i^2\right)^2 + \sum_{k=1}^{m} \frac{2}{\pi_k'^2} x^{2\alpha+1} \left(x^2 - \xi_k^2\right) \prod_{\substack{i=1 \\ i \neq k}}^{m} \left(x^2 - \xi_i^2\right)^2. \end{aligned} \tag{11.14}$$

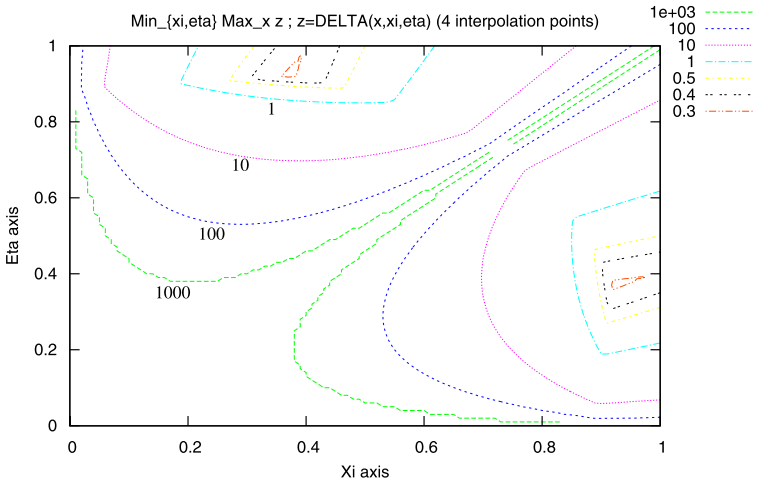


Fig. 11.1 Contour plot of function $z(\xi, \eta)$ of (11.16) for $n+1=4$: the plot is symmetrical w.r.t. the bisecting line $\eta = \xi$, and one location of the minimum is $\xi = 0.351$, $\eta = 0.926$

All the terms in $\Delta(x)$ are products of positive factors that are monotone-increasing with x. Consequently, the maximum is achieved at $x = 1$. But

$$\Delta(1) = \frac{\alpha}{\pi_0'^2}\prod_{i=1}^{m}\left(1-\xi_i^2\right)^2 + \sum_{k=1}^{m}\frac{2}{\pi_k'^2}\left(1-\xi_k^2\right)\prod_{\substack{i=1\\ i\neq k}}^{m}\left(1-\xi_i^2\right)^2.$$

This gives

$$\delta_1 = \max_{x\in[\xi_m,1]}\Delta(x) = \Delta(1) = \prod_{i=1}^{m}\left(1-\xi_i^2\right)^2\left[\frac{\alpha}{\pi_0'^2} + 2\sum_{k=1}^{m}\frac{1}{\pi_k'^2\left(1-\xi_k^2\right)}\right]. \quad (11.15)$$

11.4.3 Special Cases

For $n = 0, 1, 2$, the formal treatment is given in [1].

For $n = 3$, the four interpolation points form a symmetrical set $\{\pm\xi, \pm\eta\}$. Hence the optimization is made over two parameters ξ and η. Thus, the function

$$z(\eta, \eta) = \max_{x\in[0,1]}\Delta(x) \quad (11.16)$$

is defined discretely. The iso-value contours of this function are indicated in Fig. 11.1, which permits after refinement to identify the optimum $\xi \approx 0.351$ and $\eta \approx 0.926$.

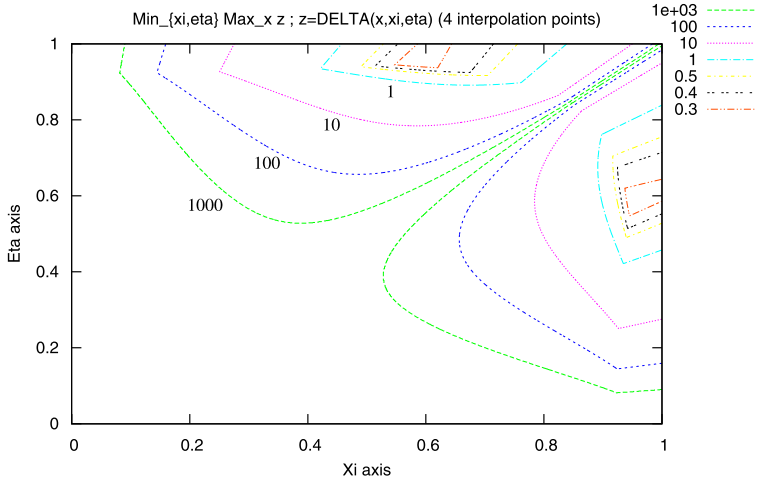


Fig. 11.2 Contour plot of function $z(\xi, \eta)$ of (11.16) for $n + 1 = 5$: the plot is symmetrical w.r.t. the bisecting line $\eta = \xi$, and one location of the minimum is $\xi = 0.571$, $\eta = 0.948$

For $n = 4$, the five interpolation points form a symmetrical set $0, \{\pm\xi, \pm\eta\}$. Hence the optimization is again made over two parameters ξ and η. Thus, the function $z(\eta, \eta)$ is again defined by (11.16) and evaluated discretely. The iso-value contours of this function are indicated in Fig. 11.2, which permits after refinement to identify the optimum $\xi \approx 0.571$ and $\eta \approx 0.948$.

11.4.4 General Results ($n > 4$)

The min-max-Δ problem has been solved by either analytical or numerical means for values of n in the range from 0 to 40. The results are collected in Table 11.1 in which the first column indicates the number of interpolation points $n + 1$, the second gives the definition of the Chebyshev points $\xi^\star$ ($n \leq 4$), the third provides the definition of the optimal distribution $\bar{\xi}$, and the fourth a comparison of performance by giving, when available, the values of

1. $\bar{\Delta} = \max_x \Delta(x, \bar{\xi})$, the upper bound on $\Delta(x)$ corresponding to the optimal distribution $\xi = \bar{\xi}$ of interpolation points;
2. $\Delta^\star = \max_x \Delta(x, \xi^\star)$, the upper bound on $\Delta(x)$ corresponding to the approximately optimal distribution $\xi = \xi^\star$ of interpolation points (the Chebyshev distribution);
3. $\Delta^u = \max_x \Delta(x, \xi^u)$, the upper bound on $\Delta(x)$ corresponding to the uniform distribution $\xi = \xi^u$ of interpolation points.

The analytical results are related to the cases for which $n \leq 4$, and have been outlined in a previous subsection.

Table 11.1 Variation of the criterion $\max_x \Delta(x,\xi)$, related to Hermitian interpolation with uncertain derivatives, for different choices of the set $\xi = \{\xi_i\}$, $i = 1,\ldots,n$, of interpolation points in $[-1,1]$, and different degrees, $2n+1$; $\bar{\Delta} = \max_x \Delta(x,\bar{\xi})$, $\Delta^\star = \max_x \Delta(x,\xi^\star)$ and $\Delta^u = \max_x \Delta(x,\xi^u)$, where $\xi_i^u = -1 + \frac{2}{n-1}(i-1)$, $i = 1,\ldots,n$

Number interpol. pts.: $n+1$ Degree of interpol.: $2n+1$	Chebyshev points: $\xi^\star$	$\bar{\xi} =$ $\arg\min_\xi \max_x \Delta$	Performance $\bar{\Delta}$	Δ^u	$\Delta^\star$
1	0	0	1		1
1				1	
2	$\pm\frac{1}{\sqrt{2}} \doteq \pm 0.7071$	± 0.7549	0.3774		0.5
3				0.5	
3	0	0	0.3258		0.3333
5	$\pm\frac{\sqrt{3}}{2} \doteq \pm 0.8660$	± 0.8677		0.3755	
4	$\pm\sqrt{\frac{1}{2}-\frac{1}{\sqrt{8}}} \doteq \pm 0.3827$	± 0.351	0.282		0.299
7	$\pm\sqrt{\frac{1}{2}+\frac{1}{\sqrt{8}}} \doteq \pm 0.9239$	± 0.926		0.439	
5	0	0	0.249		0.262
9	$\pm\sqrt{\frac{5-\sqrt{5}}{8}} \doteq \pm 0.5878$ $\pm\sqrt{\frac{5+\sqrt{5}}{8}} \doteq \pm 0.9511$	± 0.571 ± 0.948		0.652	
10			0.164		0.179
19				39.	
11			0.154		0.167
21				111.	
20			0.103		0.112
39				3.9×10^6	
21			0.100		0.108
41				1.3×10^7	

For $n+1 \geq 10$, the distribution $\bar{\xi}$ (not given here) has been identified by a numerical minimization realized by a particle-swarm (PSO) algorithm. The table indicates the corresponding values of $\bar{\Delta}$.

From these results, one observes that the upper bound $\bar{\Delta}$ achieved when the distribution of interpolation points is optimized, is not only bounded, but it even diminishes with increasing n. The Chebyshev distribution has an almost equivalent performance. Inversely, the uniform distribution yields a value of the upper bound Δ^u that is unbounded with n. In conclusion, using the Chebyshev distribution, which is known explicitly, is highly recommended in practice.

11.5 Generalized Hermitian Interpolation

In this section, we generalize the notions introduced in the first three to the situation where one wishes to construct a(the) low(est)-degree polynomial interpolant of the values, as well as the derivatives up to order, say p ($p \in \mathbb{N}$), of a given smooth function $f(x)$ over $[-1, 1]$. The interpolation points are again denoted $\{x_i\}_{i=0,1,\ldots,n}$, and we use the notation

$$y_i^{(k)} = f^{(k)}(x_i), \quad k = 0, 1, \ldots, p; \; i = 0, 1, \ldots, n.$$

The interpolation polynomial is denoted $\mathsf{H}_{n,p}(x)$ and it is now constrained to the following $(p+1)(n+1)$ interpolation conditions:

$$\forall k \in \{0, 1, \ldots, p\}, \quad \forall i \in \{0, 1, \ldots, n\} \; : \; \mathsf{H}_{n,p}^{(k)}(x_i) = y_i^{(k)}. \tag{11.17}$$

We associate such kind of interpolation with the expression "generalized Hermitian interpolation".

11.5.1 Existence and Uniqueness

We first establish existence and uniqueness by the following:

Theorem 11.1 *There exists a unique polynomial* $\mathsf{H}_{n,p}(x)$ *of degree at most equal to* $(p+1)(n+1) - 1$ *satisfying the generalized interpolation conditions* (11.17).

Proof By recurrence on p. For $p = 0$, the generalized Hermitian interpolation reduces to the classical Lagrange interpolation, whose solution is indeed unique among polynomials of degree at most equal to $(p+1)(n+1) - 1 = n$:

$$\mathsf{H}_{n,0}(x) = P_n(x).$$

For $p \geq 1$, assume $\mathsf{H}_{n,p-1}(x)$ exists and is unique among polynomials of degree at most equal to $p(n+1) - 1$. This polynomial, by assumption, satisfies the following interpolation conditions:

$$\forall k \in \{0, 1, \ldots, p-1\}, \quad \forall i \in \{0, 1, \ldots, n\} \; : \; \mathsf{H}_{n,p-1}^{(k)}(x_i) = y_i^{(k)}. \tag{11.18}$$

Hence, by seeking $\mathsf{H}_{n,p}(x)$ in the form

$$\mathsf{H}_{n,p}(x) = \mathsf{H}_{n,p-1}(x) + R(x),$$

one finds that $R(x)$ should be of degree at most equal to $(p+1)(n+1) - 1$ and satisfy

$$\forall k \in \{0, 1, \ldots, p-1\}, \quad \forall i \in \{0, 1, \ldots, n\} \; : \; R^{(k)}(x_i) = 0 \tag{11.19}$$

and

$$\forall i \in \{0, 1, \ldots, n\} \ : \ R^{(p)}(x_i) = y_i^{(p)} - \mathsf{H}_{n,p-1}^{(p)}(x_i). \tag{11.20}$$

Now, (11.19) is equivalent to saying that $R(x)$ is of the form

$$R(x) = \prod_{i=0}^{n} (x - x_i)^p \cdot Q(x) = \pi(x)^p Q(x)$$

for some quotient $Q(x)$. Then, the derivative of order p of $R(x)$ at $x = x_i$ is calculated by the Leibniz formula applied to the product $u(x)v(x)$ where

$$u(x) = (x - x_i)^p, \qquad v(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} (x - x_j)^p \cdot Q(x).$$

This gives

$$R^{(p)}(x_i) = \sum_{k=0}^{p} \binom{p}{k} u^{(k)}(x_i)\, v^{(p-k)}(x_i).$$

But, $u^{(k)}(x_i) = 0$ for all k except $k = p$ yielding

$$R^{(p)}(x_i) = p!\, v(x_i) = p! \prod_{\substack{j=0 \\ j \neq i}}^{n} (x_i - x_j)^p \, Q(x_i) = p!\, \pi'(x_i)^p \, Q(x_i).$$

Thus, all the interpolation conditions are satisfied iff the polynomial $Q(x)$ fits the following interpolation conditions:

$$\forall i \in \{0, 1, \ldots, n\} \ : \ Q(x_i) = Q_i = \frac{R^{(p)}(x_i)}{p!\pi'(x_i)^p} = \frac{y_i^{(p)} - \mathsf{H}_{n,p-1}^{(p)}(x_i)}{p!\pi'(x_i)^p}.$$

Therefore, solutions exist, and the lowest-degree solution is uniquely obtained when $Q(x)$ is the Lagrange interpolation polynomial associated with the above function values. This polynomial is of degree at most equal to n. Hence, $R(x)$ and $\mathsf{H}_{n,p}(x)$ are of degree at most equal to $p(n+1) + n = (p+1)(n+1) - 1$. □

11.5.2 Interpolation Error and Best Approximation

We have the following:

Theorem 11.2 (Interpolation Error Associated with the Generalized Hermitian Interpolant) *Assuming that* $f \in \mathsf{C}^{(p+1)(n+1)}([-1,1])$, *we have*

$$\forall x \in [-1,1],\ \exists \xi \in [-1,1]\ :\ f(x) = \mathsf{H}_{n,p}(x) + \pi(x)^{p+1} \frac{f^{([(p+1)(n+1)])}(\xi)}{[(p+1)(n+1)]!}.$$

Proof Let $x \in [-1,1]$ be fixed. If $x = x_i$ for some $i \in \{0,1,\ldots,n\}$, $f(x) = \mathsf{H}_{n,p}(x)$ and $\pi(x) = 0$, and the statement is trivial. Hence, assume now otherwise that $x \neq x_i$ for any $i \in \{0,1,\ldots,n\}$. Then, define the constant

$$\gamma = \frac{f(x) - \mathsf{H}_{n,p}(x)}{\pi(x)^{p+1}}$$

so that

$$f(x) = \mathsf{H}_{n,p}(x) + \gamma \pi(x)^{p+1}.$$

Now using the symbol t for the independent variable, one considers the function

$$\theta(t) = f(t) - \mathsf{H}_{n,p}(t) - \gamma \pi(t)^{p+1}.$$

By virtue of the interpolation conditions satisfied by the polynomial $\mathsf{H}_{n,p}(t)$,

$$\forall k \in \{0,1,\ldots,p\}, \quad \forall i \in \{0,1,\ldots,n\}\ :\ \theta^{(k)}(x_i) = 0 \tag{11.21}$$

but, additionally, by the choice of the constant γ, we also have

$$\theta(x) = 0.$$

This makes $n+2$ distinct zeroes for $\theta(x) : x_0, x_1, \ldots, x_n$ and x. Thus, by application of Rolle's theorem in each of the $n+1$ consecutive intervals that these $n+2$ points once arranged in an increasing order define, a zero of $\theta'(t)$ exists, yielding $n+1$ distinct zeroes for $\theta'(t)$, to which (11.21) adds $n+1$ distinct and different ones, for a total of $2(n+1) = 2n+2$ zeroes. Strictly between these, one finds $2(n+1)-1$ zeroes of $\theta''(t)$ to which (11.21) adds $n+1$ distinct and different ones, for a total of $3(n+1)-1 = 3n+2$ zeroes. Thus, for every new derivative, we find one less zero in every subinterval, but $n+1$ more by virtue of (11.21), for a total of n more, and this as long as (11.21) applies. Hence we get that $\theta^{(p)}(t)$ admits at least $(p+1)n+2$ distinct zeroes. For derivatives of higher order, the number of zeroes is one less for every new one; hence, $(p+1)n+1$ for $\theta^{(p+1)}(t)$, and so on. We finally get that $\theta^{([p+(p+1)n+1])}(t) = \theta^{([(p+1)(n+1)])}(t)$ admits at least one zero ξ, that is

$$0 = f^{([(p+1)(n+1)])}(\xi) - \gamma\big[(p+1)(n+1)\big]!$$

because $\mathsf{H}^{([(p+1)(n+1)])}(\xi) = 0$ since the degree of $\mathsf{H}_{n,p}(t)$ is at most equal to $(p+1)(n+1)-1$, and the conclusion follows. □

As a consequence of this result, it is clear that the best generalized Hermitian approximation is achieved by the Chebyshev distribution of interpolation points again.

11.5.3 Best Inexact Generalized Hermitian Interpolation

Now suppose that all the data on f and its successive derivatives are exact, except for the derivatives of the highest order, $\{y_i^{(p)}\}$ that are subject to uncertainties $\{\epsilon_i\}_{i=0,1,\ldots,n}$. Then, the uncertainties on the values $\{Q_i\}_{i=0,1,\ldots,n}$ of the quotient $Q(x)$ are the following:

$$\delta Q_i = \frac{\epsilon_i}{p!\pi'(x_i)^p};$$

on the quotient itself the following:

$$\delta Q(x) = \sum_{i=0}^{n} \frac{\epsilon_i}{p!\pi'(x_i)^p} \mathsf{L}_i(x) = \pi(x) \sum_{i=0}^{n} \frac{\epsilon_i}{p!\pi'(x_i)^{p+1}(x-x_i)};$$

and, finally, the uncertainty on the generalized Hermitian interpolant $\mathsf{H}_{n,p}(x)$ the following:

$$\delta \mathsf{H}_{n,p}(x) = \pi(x)^{p+1} \sum_{i=0}^{n} \frac{\epsilon_i}{p!\pi'(x_i)^{p+1}(x-x_i)}.$$

In conclusion, for situations in which the uncertainties $\{\epsilon_i\}_{i=0,1,\ldots,n}$ are bounded by the same number ϵ, the criterion that one should consider to conduct the min-max optimization of the interpolation points $\{x_i\}_{i=0,1,\ldots,n}$ is now the following one to replace the former $\Delta(x)$:

$$\Delta^{(p)}(x) = \left|\pi(x)\right|^{p+1} \sum_{i=0}^{n} \frac{1}{p!|\pi'(x_i)|^{p+1}|x-x_i|} \tag{11.22}$$

or, equivalently,

$$\sqrt[p+1]{\Delta^{(p)}(x)} = \left|\pi(x)\right| \sqrt[p+1]{\sum_{i=0}^{n} \frac{1}{p!|\pi'(x_i)|^{p+1}|x-x_i|}}.$$

We note that this expression is a homogeneous function of $\pi(x)$ of degree 0.

We conjecture that the variations of the above criterion, as $p \to \infty$, are dominated by those of the factor $|\pi(x)|$. Hence, in this limit, the optimal distribution of interpolation points should approach the Chebyshev distribution.

11.5.4 Overall Bound on the Approximation Error

The quantity $\epsilon\Delta^{(p)}(x)$ is an absolute bound on the error committed in the computation of the generalized Hermitian interpolant based on function and derivative

values in presence of uncertainties on the derivatives of the highest order, p, only, when these are uniformly bounded by ϵ:

$$\forall x \in [-1, 1] \; : \; \left|\delta \mathsf{H}_{n,p}(x)\right| = \left|\bar{\mathsf{H}}_{n,p}(x) - \mathsf{H}_{n,p}(x)\right| \leq \epsilon \Delta^{(p)}(x),$$

where $\bar{\mathsf{H}}_{n,p}(x)$ represents the actually computed approximation.

On the other hand, the interpolation error is the difference between the actual function value, $f(x)$, and the "true" interpolant, $\mathsf{H}_{n,p}(x)$, that could be computed if all function and derivative information was known. The interpolation error satisfies

$$\forall x \in [-1, 1] \; : \; \left|f(x) - \mathsf{H}_{n,p}(x)\right| = \left|\pi(x)^{p+1} \frac{f^{([(p+1)(n+1)])}(\xi)}{[(p+1)(n+1)]!}\right| \leq \mu_{n,p} \left|\pi(x)\right|^{p+1},$$

where one has let

$$\mu_{n,p} = \max_{x \in [-1,1]} \left| \frac{f^{([(p+1)(n+1)])}(x)}{[(p+1)(n+1)]!} \right|.$$

Consequently, we have

$$\forall x \in [-1, 1] \; : \; \left|f(x) - \bar{\mathsf{H}}_{n,p}(x)\right| \leq \mu_{n,p} \left|\pi(x)\right|^{p+1} + \epsilon \Delta^{(p)}(x). \tag{11.23}$$

Now, examining the precise expression for $\Delta^{(p)}(x)$, that is (11.22), we see that the ratio of the second term to the first on the right of the above inequality is equal to

$$\frac{\epsilon}{\mu_{n,p}} \sum_{i=0}^{n} \frac{1}{p! |\pi'(x_i)|^{p+1} |x - x_i|}.$$

For given n and p, this expression is unbounded in x. Thus, (the bound on) the error is inevitably degraded in the order of magnitude due to presence of uncertainties.

However, the actual dilemma of interest is somewhat different. It is the following: given the values $\{y_i, y_i', \ldots, y_i^{(p-1)}\}$, $0 \leq i \leq n$, and correspondingly, approximations of the higher derivative $\{y_i^{(p)}\}$, which of the following two interpolants is (guaranteed to be) more accurate:

1. the Hermitian interpolant of the sole exact values: $\{y_i, y_i', \ldots, y_i^{(p-1)}\}$, $0 \leq i \leq n$, or
2. the Hermitian interpolant of the entire data set?

The first interpolant differs from $f(x)$ by the sole interpolation error, $\mu_{n,p-1}|\pi(x)|^p$. The second interpolant is associated with the higher-order interpolation error, $\mu_{n,p}|\pi(x)|^{p+1}$, but is subject to the uncertainty term $\epsilon \Delta^{(p)}(x)$, which is dominant, as we have just seen. Thus, the decision of whether to include derivatives or not should be guided by the ratio of the uncertainty term, $\epsilon \Delta^{(p)}(x)$, to the lower interpolation error, $\mu_{n,p-1}|\pi(x)|^p$. This ratio is equal to

$$\frac{\epsilon}{\mu_{n,p-1}} \left|\pi(x)\right| \sum_{i=0}^{n} \frac{1}{p! |\pi'(x_i)|^{p+1} |x - x_i|}$$

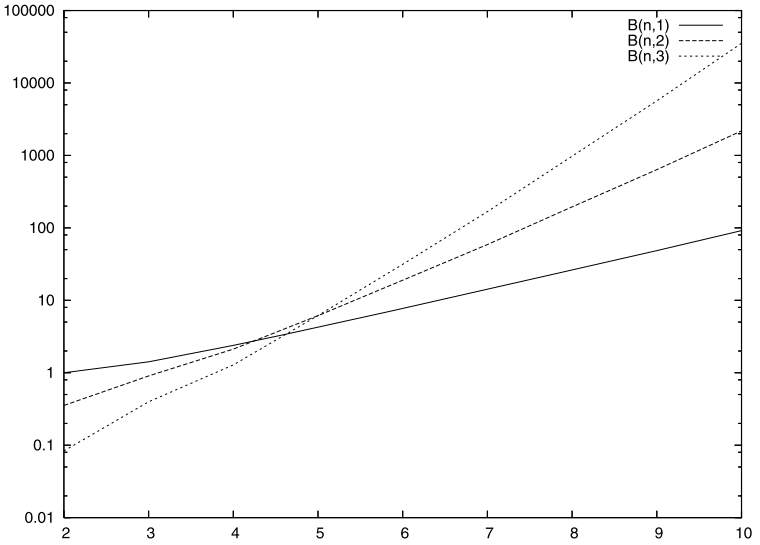


Fig. 11.3 Coefficient $B_{n,p}$ as a function of n for $p = 1, 2$ and 3

and it admits the bound

$$\frac{\epsilon B_{n,p}}{\mu_{n,p-1}}, \tag{11.24}$$

where the bound

$$B_{n,p} = \max_{x\in[-1,1]} \left|\pi(x)\right| \sum_{i=0}^{n} \frac{1}{p!|\pi'(x_i)|^{p+1}|x - x_i|} \tag{11.25}$$

exists since, in the above, the function over which the max applies is piecewise polynomial for fixed n and p.

Hermitian interpolation is definitely preferable whenever the expression in (11.24) is less than 1. This criterion permits us to identify trends as ϵ, n and p vary, but is not very practical in general since the factors ϵ and $\mu_{n,p-1}$ are problem-dependent and out of control. The variation with n of the bound $B_{n,p}$ has been plotted in Fig. 11.3 for $p = 1, 2$ and 3. Visibly, the bound $B_{n,p}$ can be large unless p and n are small. Therefore, unsurprisingly, unless n and p, as well as the uncertainty level ϵ, are small enough, the criterion in (11.24) is larger than 1, and the interpolant of the sole exactly known values is likely to be the more accurate one.

To appreciate this in a practical case, we have considered the case of the interpolation of the function

$$f(x) = f_\lambda(x) = \frac{1}{1 + \lambda x^2}$$

over the interval $[-1, 1]$ for $p = 0$ (Lagrange interpolation) and $p = 1$ (Hermitian interpolation). This smooth function is bounded by 1, and its maximum derivative

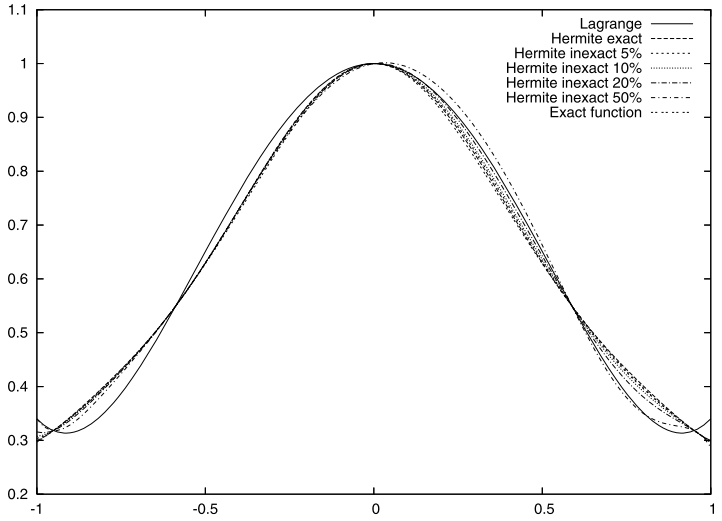


Fig. 11.4 Case $\lambda = 64/27$ ($\max_x |f_\lambda(x)| = \max_x |f'_\lambda(x)| = 1$); function $f_\lambda(x)$ and various interpolation polynomials ($n = 5$)

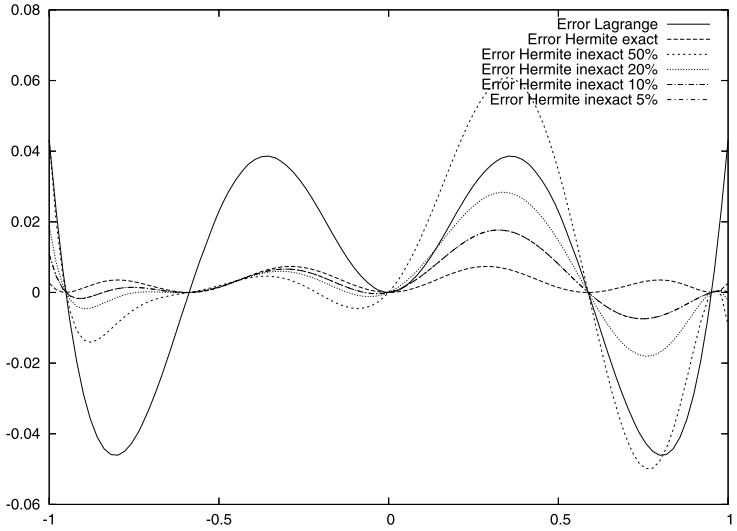


Fig. 11.5 Case $\lambda = 64/27$ ($\max_x |f_\lambda(x)| = \max_x |f'_\lambda(x)| = 1$); error distribution associated with the various interpolation polynomials ($n = 5$)

increases with λ. For $\lambda = 64/27$, this maximum is equal to 1. For $\lambda = 256/27$, this maximum is equal to 2.

In the first experiment (Figs. 11.4 and 11.5), $\lambda = 64/27$ and $n = 5$. The Lagrange interpolant is fairly inaccurate, mostly near the endpoints. Thus the error distribution

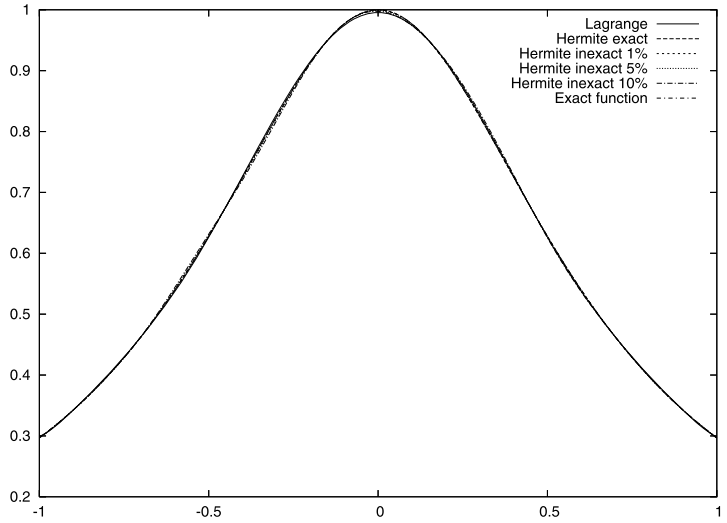


Fig. 11.6 Case $\lambda = 64/27$ ($\max_x |f_\lambda(x)| = \max_x |f'_\lambda(x)| = 1$); function $f_\lambda(x)$ and various interpolation polynomials ($n = 10$)

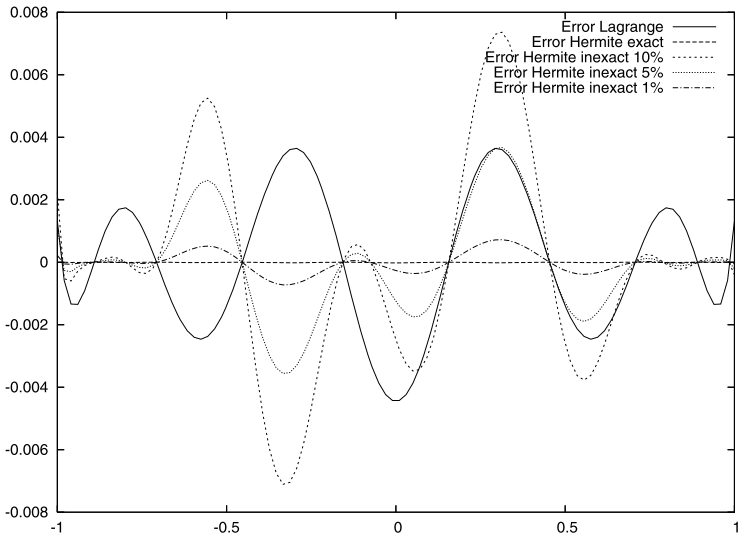


Fig. 11.7 Case $\lambda = 64/27$ ($\max_x |f_\lambda(x)| = \max_x |f'_\lambda(x)| = 1$); error distribution associated with the various interpolation polynomials ($n = 10$)

indicates that the approximate Hermitian interpolant is preferable even for a fairly high level of uncertainty on the derivatives (20 % is acceptable).

In the second experiment (Figs. 11.6 and 11.7), the interpolated function is the same, but the number n is doubled ($n = 10$). Consequently, the Lagrange interpolant

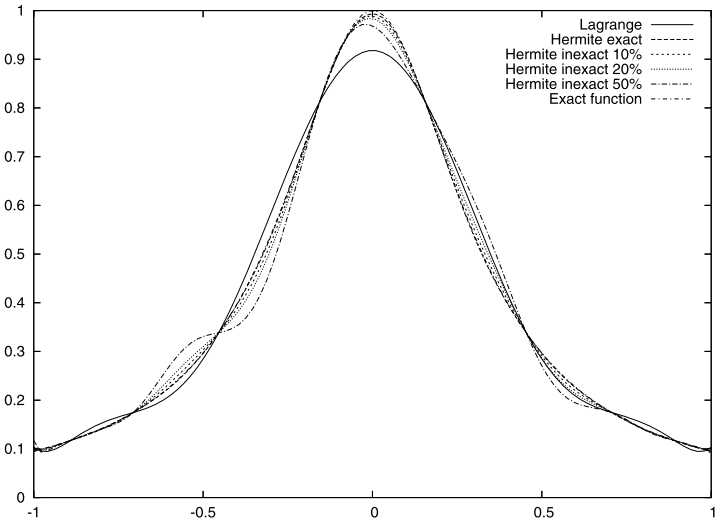


Fig. 11.8 Case $\lambda = 256/27$ ($\max_x |f_\lambda(x)| = 1$; $\max_x |f'_\lambda(x)| = 2$); function $f_\lambda(x)$ and various interpolation polynomials ($n = 10$)

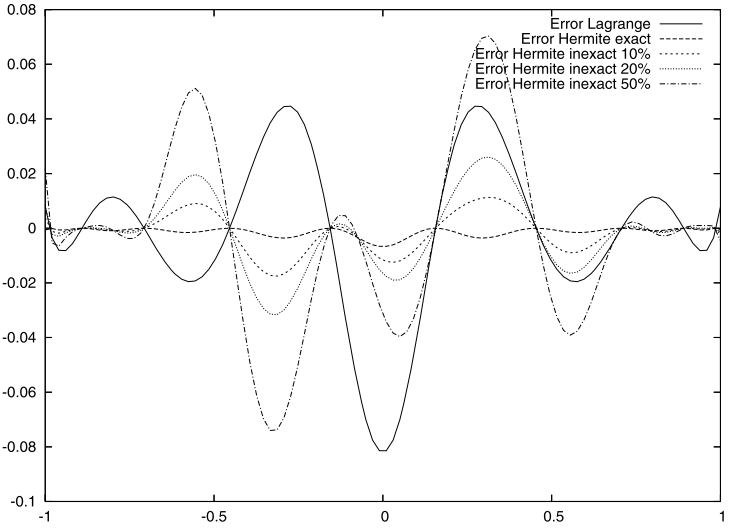


Fig. 11.9 Case $\lambda = 256/27$ ($\max_x |f_\lambda(x)| = 1$; $\max_x |f'_\lambda(x)| = 2$); error distribution associated with the various interpolation polynomials ($n = 10$)

of the sole exact function values is very accurate. The approximate Hermitian interpolant can only surpass it if the level of uncertainty on the derivatives is small (less than 5 %).

Lastly, with the same number of interpolation points ($n = 10$), we have considered the case of a function with larger derivatives ($\lambda = 256/27$). As a result (see Figs. 11.8 and 11.9), the accuracy of the Lagrange interpolation has been severely degraded. Then again, the approximate Hermitian interpolation is found superior for higher levels of uncertainty in the derivatives (the switch is between 20 % and 50 %).

11.6 Conclusions

Recalling that the Chebyshev distribution of interpolation points is optimal w.r.t. the minimization of the (known bound on the) interpolation error, we have proposed an alternate criterion to be subject to the min-max optimization. The new criterion to be minimized aims at reducing the sensitivity of the Hermitian interpolant of function values and derivatives, to uncertainties assumed to be present in the derivatives only. We have found by analytical developments and numerical experiments that the Chebyshev distribution is close to be optimum w.r.t. this new criterion also, thus giving the stability of the corresponding approximation a somewhat larger sense.

We have also considered the generalized Hermitian interpolation problem in which the derivatives up to some order p ($p > 1$) are fitted. For this problem we have derived the existence and uniqueness result, as well as the expression of the interpolation error, and also the definition that one could use for the criterion to be subject to the min-max optimization to reduce the sensitivity of the interpolant to uncertainties in the derivatives of the highest order. We conjectured from the derived expression that the corresponding optimal distribution of interpolation points converges to the Chebyshev distribution as $p \to \infty$.

Lastly, we have made actual interpolation experiments in cases of a function bounded by 1, whose derivative is either bounded by 1 or 2. These experiments have confirmed that the approximate Hermitian interpolant was superior to the Lagrange interpolant of the sole exact function values, when the uncertainty on the derivatives is below a certain critical value which decreases when n is increased.

In perspective, we intend to examine a much more complicated case in which the meta-model depends nonlinearly on the adjustable parameters, by means of semi-formal or numerical analysis tools.

References

1. Bompard M, Désidéri J-A, Peter J (2010) Best Hermitian interpolation in presence of uncertainties. INRIA Research Report RR-7422, Oct. http://hal.inria.fr/inria-00526558/en/
2. Conte SD, de Boor C (1972) Elementary numerical analysis: an algorithmic approach, 2nd edn. McGraw-Hill, New York
3. Jones D (2001) A taxonomy of global optimization methods based on response surfaces. J Glob Optim 21(4):345–383

4. Pearson CE (ed) (1990) Handbook of applied mathematics: selected results and methods, 2nd edn. Van Nostrand Reinhold, New York
5. Quarteroni A, Sacco R, Saleri F (2007) Numerical mathematics, 2nd edn. Texts in applied mathematics, vol 37. Springer, Berlin
6. Sacks J, Welch W, Mitchell T, Wynn H (1989) Design and analysis of computer experiments. Stat Sci 4(4):409–435

Chapter 12
Inversion of the Heat Equation by a Block Based Algorithm Using Spline Wavelet Packets

Amir Averbuch, Pekka Neittaanmäki, and Valery Zheludev

Abstract We present a robust algorithm starting from 1D or 2D discrete noised data to approximately invert the heat equation, which is an ill-conditioned problem. Relative contributions of the coherent structure and the noise in different frequency bands of the available data are different. We propose to solve the inversion problem separately in different frequency bands by methods similar to the Tikhonov regularization. This separation is achieved by using spline wavelet packets. The solutions are derived as linear combinations of those wavelet packets.

12.1 Introduction

The problems are formulated as follows: Let the functions $f(x) \in C^2(\mathbb{R}^1)$ and $f(x,y) \in C^2(\mathbb{R}^2)$ be compactly supported. Denote by $\mathbf{U}_t^1$ and by $\mathbf{U}_t^2$ the linear operators such that $\mathbf{U}_t^1 f(x) = g(x,t)$ and $\mathbf{U}_t^2 f(x,y) = g(x,y,t)$ where $g(x,t)$ and $g(x,y,t)$ are the solutions of the heat equations with the initial conditions $f(x)$ and $f(x,y)$, respectively:

$$\begin{aligned} \frac{\partial g(x,t)}{\partial t} &= g''_x(x,t), & g(x,0) &= f(x), \\ \frac{\partial g(x,y,t)}{\partial t} &= g''_x(x,y,t) + g''_y(x,y,t), & g(x,y,0) &= f(x,y). \end{aligned} \tag{12.1}$$

A. Averbuch (✉) · V. Zheludev
School of Computer Science, Tel Aviv University, P.O. Box 39040, Tel Aviv 69978, Israel
e-mail: amir@math.tau.ac.il

V. Zheludev
e-mail: zhel@post.tau.ac.il

A. Averbuch · P. Neittaanmäki · V. Zheludev
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland

P. Neittaanmäki
e-mail: pekka.neittaanmaki@mit.jyu.fi

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_12,

Problem 12.1 Let t be a fixed time parameter. Given $g(x,t) = \mathbf{U}_t^1 f(x)$, find $f(x)$.

Problem 12.2 Let t be a fixed time parameter. Given $g(x,y,t) = \mathbf{U}_t^2 f(x,y)$, find $f(x,y)$.

For brevity, we concentrate on Problem 12.1. Extension to the 2D case is straightforward. The problem has explicit theoretical solutions [6]. We assume that the initial temperature distribution $f(x)$ is a T-periodic function. Consequently, $g(x,t) = \mathbf{U}_t^1 f(x)$ is T-periodic as well. These functions can be expanded into the following Fourier series:

$$f(x) = \frac{1}{T}\sum_{n\in\mathbb{Z}} f_n e^{2\pi i n x/T}, \qquad g(x,t) = \frac{1}{T}\sum_{n\in\mathbb{Z}} g_n(t) e^{2\pi i n x/T}, \qquad g_n(0) = f_n.$$

If we know the function $g(x,t)$ at some fixed t then

$$f_n = g_n(t)e^{t(2\pi n/T)^2} \Longrightarrow f(x) = \frac{1}{T}\sum_{n\in\mathbb{Z}} g_n(t) e^{t(2\pi n/T)^2} e^{2\pi i n x/T}. \tag{12.2}$$

In real life, the function $g(x,t)$ is known up to some errors, modeled as $\tilde{g}(x,t) = g(x,t) + e(x)$. Generally, there is no reason to assume that the Fourier coefficients of the error tend to zero faster than $e^{-n^2 t}$ (if they tend to zero at all). Therefore, according to (12.2), application of the direct inversion to the available data $\tilde{g}(x,t)$ results in an unstable solution. However, as the magnitude of the error function $|e(x)|$ becomes smaller, the function $\tilde{f}(x)$ can comprise strong high-frequency components, which do not exist in the original function $f(x)$. Therefore, a regularization, which provides a stability to the solution at the expense of deviation from the available data $\tilde{g}(x,t)$, is needed.

Typically, the data function $g(x,t)$ is sampled on a grid $\{x[k]\}$ and the samples are corrupted by a random noise, while the sought-after initial temperature distribution $f(x)$ is continuous. Therefore, it is reasonable to design approximated solutions as splines. Splines bridge the gap between the discrete input data and the continuous solution. To take into account different relative shares of the coherent signal and the noise in different frequency components of the available data, we propose to solve the inversion problem separately in different frequency bands. This approach significantly extends the adaptation abilities and the robustness of the method. Practically, this scheme is implemented via the application of the orthonormal spline wavelet packets, which are constructed by using the Spline Harmonic Analysis (SHA) framework. The wavelet packet transform splits the frequency domain of a signal into a set of bands whose overlap is minimal.

12.2 Elements of SHA

We briefly outline the basics of the SHA techniques. A detailed description is given, for example, in [2, 3].

We assume that $N = 2^j$, $j \in \mathbb{N}$ and $p = 2r > 0$ is an even integer. The space of N-periodic splines of even order $p = 2r$, which have nodes on the grid $\{k\}$, is denoted by ${}^p\mathscr{S}$. A basis in ${}^p\mathscr{S}$ is formed by translations of the centered periodic B-spline $B^p(x)$:

$$S^p(x) = \sum_{k=0}^{N-1} q[k] B^p(x-k) \in {}^p\mathscr{S}.$$

The B-spline $B^p(x)$ belongs to the space C^{p-2}. The circular convolution $B^p \star B^r(x) = B^{p+r}(x) \Longrightarrow S_1^p \star S_2^r(x) = S_3^{p+r}(x) \in {}^{p+r}\mathscr{S}$. Thus, the circular convolution of two periodic splines is a spline. Therefore, splines are a proper tool for dealing with convolution-type problems where inversion of the heat equation belongs to.

There exist orthogonal bases in ${}^p\mathscr{S}$ which resemble the Fourier basis. Denote $\omega \stackrel{\text{def}}{=} e^{2\pi i/N}$. The orthogonal basis of the space ${}^p\mathscr{S}$ is constituted by *exponential splines*

$$\beta^p[n](x) \stackrel{\text{def}}{=} \sum_{k=0}^{N} \omega^{-nk} B^p(x+k), \quad n = 0, \ldots, N-1.$$

The following representation holds:

$$S^p(x) = \sum_{k=0}^{N-1} q[k] B^p(x-k) = \frac{1}{N} \sum_{n=0}^{N-1} \widehat{q}[n] \beta^p[n](x).$$

Here $\widehat{q}[n] = \sum_{k=0}^{N} \omega^{-nk} q[k]$ is the discrete Fourier transform (DFT) of the coefficients $\{q[k]\}$. For further use, we single out the sequence

$$u^p[n] \stackrel{\text{def}}{=} \beta^p[n](0) = \sum_{k=0}^{N-1} \omega^{-nk} B^p(k), \tag{12.3}$$

which is the DFT of the sampled B-spline. The sequences $u^p[n]$ are N-periodic and strictly positive. The norms of the exponential splines are $\|\beta^p[n]\| = \sqrt{N u^{2p}[n]}$. Thus, the splines,

$$\gamma^p[n](x) \stackrel{\text{def}}{=} \frac{\beta^p[n](x)}{\|\beta^p[n]\|} = \frac{\beta^p[n](x)}{\sqrt{N u^{2p}[n]}}, \quad n = 0, \ldots, N-1,$$

form an orthonormal basis of $\mathscr{S}^p$. The spline $S^p(x) \in {}^p\mathscr{S}$ is represented by

$$S^p(x) = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} \sigma[n] \gamma^p[n](x), \quad \sigma[n] = \sqrt{u[n]^{2p}}\, \hat{q}[n]. \tag{12.4}$$

This expansion imposes a specific form of the SHA methodology onto the spline space, where the splines $\{\gamma^p[n](x)\}_{n=0}^{N-1}$ act as harmonics and the coordinates

$\{\sigma[n]\}$, $n = 0, \ldots, N-1$, which we refer to as to the SHA spectrum of the spline $S^p(x)$. They act as the Fourier coefficients. Many operations on splines are significantly simplified [2, 3]. Denote by δ^2 the second central difference: $\delta^2 f(x) = f(x+1) - 2f(x) + f(x-1)$ and

$$w[n] \stackrel{\text{def}}{=} 4\sin^2\frac{\pi n}{M}, \qquad W[n] \stackrel{\text{def}}{=} \sqrt{\frac{u^{2(p-2)}[n]}{u^{2p}[n]}}\, w[n], \qquad V[n] \stackrel{\text{def}}{=} \frac{u^p[n]}{\sqrt{u^{2p}[n]}}. \tag{12.5}$$

Then

$$\delta^2 S^p(x) = -\sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} w[n]\sigma[n]\gamma^p[n](x), \qquad S^p(k) = \frac{1}{N}\sum_{n=0}^{N-1} \omega^{kn} V[n]\sigma[n], \tag{12.6}$$

$$S''(x) = -\sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} W[n]\sigma[n]\gamma^{p-2}[n](x), \qquad \left\|S''\right\|^2 = \frac{1}{N}\sum_{n=0}^{N-1} \left|W[n]\sigma[n]\right|^2. \tag{12.7}$$

It follows from (12.6) that, if a spline $S^p(x)$ interpolates a sequence $\mathbf{y} = \{y[k]\}$ at grid points $S^p(k) = y[k]$, then its SHA spectrum is

$$\sigma[n] = \frac{\hat{y}[n]}{V[n]}, \quad \hat{y}[n] = \sum_{k=0}^{N-1} \omega^{-kn} y[k]. \tag{12.8}$$

12.3 Global Regularized Spline Solution

We briefly outline the scheme for global solution, which is a realization of the Tikhonov regularization algorithm [7] in the space of periodic splines. A full presentation of the scheme is given in [1].

To immerse Problem 12.1 into the spline setting, $\mathbf{V}_t$ denotes the linear operator defined on the spline space ${}^p\mathscr{S}$ such that $\mathbf{V}_t S(x) = s(x,t)$, where $s(x,t)$ is the spline solution to the difference approximation of the heat equation

$$\frac{\partial s(x,t)}{\partial t} = \delta_x^2 s(x,t), \qquad s(x,0) = S(x) \tag{12.9}$$

from ${}^p\mathscr{S}$ (with respect to x).

Assume the spline $S^p(x)$ is represented by (12.4). The spline $s(x,t) \in \mathscr{S}^p$ can be represented as

$$s(x,t) = N^{-1/2} \sum_{n=0}^{N-1} \sigma[n](t)\gamma^p[n](x).$$

Using (12.6), we get

$$\sigma[n](t) = \eta[n](t)\sigma[n], \tag{12.10}$$

where $\eta[n](t) = e^{-w[n]t}$. If we know the spline $s(x,t)$ by a fixed t then the spline $S(x)$ becomes

$$S(x) = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} e^{w[n]t} \sigma[n](t) \gamma^p[n](x).$$

However, typically only the data vector $\mathbf{z} = \{z[k]\}$, $k = 0, \ldots, N-1$, is known, where $z[k] = g(k,t) + e_k$, $\mathbf{e} = \{e_k\}$ are the measurement errors, which we assume to be white noise.

The approximated solution to Problem 12.1 is derived as a spline

$$S_\rho(x) = \arg \min_{S \in \mathscr{S}^p} \big(\rho I(S) + E(S)\big),$$

where

$$I(S) \stackrel{\text{def}}{=} \|S''\|^2 = \frac{1}{N} \sum_{n=0}^{N-1} \big|W[n]\sigma[n]\big|^2,$$

$$E(S) \stackrel{\text{def}}{=} \sum_k \big(\mathbf{V}_t S(k) - z[k]\big)^2 = \frac{1}{N} \sum_{n=0}^{N-1} \big|V[n]\sigma[n]\eta[n](t) - \hat{z}[n]\big|^2.$$

A spline solution to the minimization problem is

$$S_\rho(x) = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} \sigma[n](\rho)\gamma^p[n](x), \quad \sigma[n](\rho) = \frac{\overline{\eta}[n](t)\hat{z}[n]V[n]}{A[n](\rho)},$$

where $\overline{\eta}[n](t)$ is the complex conjugate of $\eta[n](t)$, $A[n](\rho) \stackrel{\text{def}}{=} \rho W[n] + (\eta[n](t)V[n])^2$.

Assume we are able to estimate the variance $\text{var}(\mathbf{e}) = \varepsilon^2$ of the error vector. The regularization parameter ρ is derived from the solution of the equation

$$e(\rho) \stackrel{\text{def}}{=} E(S_\rho)/N = \frac{1}{N^2} \sum_{n=0}^{N-1} \left(\frac{\rho W[n]|\hat{z}[n]|}{A[n](\rho)} \right)^2 = \varepsilon^2. \tag{12.11}$$

The function $e(\rho)$ grows strictly monotonically as $\rho \to \infty$ and $\lim_{\rho\to\infty} e(\rho) = N^{-2}\|\mathbf{z}\|^2$. If $N^{-1}\|\mathbf{z}\| > \varepsilon$, then (12.11) has a unique solution.

The parameter ρ, which provides a trade-off between approximation and regularization, depends on the relative shares of the coherent signal and the noise in the available data. These shares are different in different frequency components of the data. We propose to solve the problems separately in different frequency bands, while the regularization parameters are to be found according to the signal-to-noise

ratio in each band. It is achieved by the application of the orthonormal spline wavelet packet transform, which splits the frequency domain of a signal into a set of bands whose overlap is minimal. The SHA framework provides tools for the design of wavelet packets and for the efficient implementation of the algorithm.

12.4 Wavelet Packets

Denote by ${}^{p}\mathcal{S}_{r,0}$, $r \in \mathbb{N}$, the space of N-periodic splines of even order p on the grid $\{2^r k\}$. In the rest of the paper $N_r \stackrel{\text{def}}{=} N/2^r$, $n_r \stackrel{\text{def}}{=} n + N_r/2$. The space ${}^{p}\mathcal{S}_{r,0}$ is an N_r-dimensional space, where a basis consists of 2^r-sample shifts of the B-splines B_r^p constructed on the grid $\{2^r k\}$. The inclusion relations between the spaces ${}^{p}\mathcal{S}_{r,0} \subset {}^{p}\mathcal{S}^{r-1,0} \subset \cdots \subset {}^{p}\mathcal{S}^{0,0} \equiv \mathcal{S}^p$ hold. Similarly to the space $\mathcal{S}^p$, the orthogonal and orthonormal bases of ${}^{p}\mathcal{S}_{r,0}$ are formed by the exponential splines

$$\beta_{r,0}^{p}[n](x) \stackrel{\text{def}}{=} \sum_{k=0}^{N_r} \omega^{-2^r nk} B^p\bigl(x + 2^r k\bigr), \qquad u_r^p[n] \stackrel{\text{def}}{=} \beta_{r,0}^{p}[n](0),$$

$$\gamma_{r,0}^{p}[n](x) \stackrel{\text{def}}{=} \frac{\beta_{r,0}^{p}[n](x)}{\sqrt{N_r u_r^{2p}[n]}}.$$

For the initial scale, we retain the notations $\gamma^p[n] \equiv \gamma_{0,0}^{p}[n](x)$, $u^p[n] \equiv u_0^p[n]$.

When it will not produce a confusion, we drop the order index $\cdot^p$.

The two-scale relation between basis splines from adjacent spaces holds to be

$$\gamma_{r,0}[n](x) = b_{r-1}[n]\gamma_{r-1,0}[n](x) + b_{r-1}[n_r]\gamma_{r-1,0}[n_r](x), \tag{12.12}$$

where

$$b_{r-1}[n] \stackrel{\text{def}}{=} \sqrt{\frac{u_{r-1}^{2p}[n]}{2u_r^{2p}[n]}}\cos^p\left(\frac{2^{r-1}\pi n}{N}\right), \quad n_r \stackrel{\text{def}}{=} n + \frac{N_r}{2}.$$

Denote by ${}^{p}\mathcal{S}_{r,1}$ the orthogonal complement to ${}^{p}\mathcal{S}_{r,0}$ in the space ${}^{p}\mathcal{S}^{r-1,0}$. An orthonormal basis in ${}^{p}\mathcal{S}_{r,1}$ contains the splines

$$\gamma_{r,1}[n](x) = \tilde{b}_{r-1}[n]\gamma_{r-1,0}[n](x) + \omega_{r-1}^{2^{r-1}n}[n_r]\gamma_{r-1,0}[n_r](x), \tag{12.13}$$

where $\tilde{b}_{r-1}[n] \stackrel{\text{def}}{=} \omega^{2^{r-1}n} b_{r-1}[n_r]$. If $r > 1$, we can apply a similar procedure to the space ${}^{p}\mathcal{S}_{r-1,1}$. As a result, we get the decomposition ${}^{p}\mathcal{S}_{r-1,1} = {}^{p}\mathcal{S}_{r,2} \oplus {}^{p}\mathcal{S}_{r,3}$. By applying the same procedure to all the derived subspaces, we decompose the spline space ${}^{p}\mathcal{S}$ into a series of orthogonal sums

$${}^{p}\mathcal{S} = {}^{p}\mathcal{S}_{1,0} \oplus {}^{p}\mathcal{S}_{1,1} = {}^{p}\mathcal{S}_{2,0} \oplus {}^{p}\mathcal{S}_{2,1} \oplus {}^{p}\mathcal{S}_{2,2} \oplus {}^{p}\mathcal{S}_{2,3} = \cdots = \bigoplus_{l=0}^{2^r-1} {}^{p}\mathcal{S}_{r,l}.$$

The orthonormal bases $\{\gamma_{r,l}[n](x)\}[n]$ of the subspaces ${}^{p}\mathscr{S}_{r,l}$ are derived iteratively by the two-scale relations using the coefficients $b_{r-1}[n]$ and $\tilde{b}_{r-1}[n]$.

Similarly to the Fourier exponentials, the exponential basis splines $\gamma_{r,l}[n](x)$ are complex-valued and are not localized in the space domain. However, their real-valued and well-localized counterparts satisfy

$$\psi_{r,l}(x) \stackrel{\text{def}}{=} \sqrt{\frac{1}{N}} \sum_{n=0}^{N_r-1} \gamma_{r,l}[n](x) \in {}^{p}\mathscr{S}_{r,l} \subset {}^{p}\mathscr{S}. \tag{12.14}$$

These splines are called the spline wavelet packets. The shifts $\{\psi_{r,l}(x-2^r k)\}$, $k = 0, \ldots, N_r - 1$, form an orthonormal basis for the space ${}^{p}\mathscr{S}_{r,l}$. Consequently, the union $\biguplus_{l=0}^{2^r-1}\{\psi_{r,l}(x-2^r k)\}$ forms an orthonormal basis for the entire space ${}^{p}\mathscr{S}$.

At the initial scale, the one-sample shifts of the splines

$$\varphi^{p}(x) \stackrel{\text{def}}{=} \psi^{p}_{0,0}(x) = N^{-1/2} \sum_{n=0}^{N-1} \gamma^{p}[n](x)$$

form an orthonormal basis.

All the spaces ${}^{p}\mathscr{S}_{r,l}$ belong to ${}^{p}\mathscr{S}$, thus, the wavelet packet $\psi_{r,l}(x)$ forms a subspace ${}^{p}\mathscr{S}_{r,l}$ and can be expanded over the orthonormal basis $\{\gamma^{p}[n](x)\}$ of $\mathscr{S}^{p}$:

$$\psi_{r,l}(x) = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} \nu_{r,l}[n]\gamma^{p}[n](x). \tag{12.15}$$

The SHA spectra $\{\nu_{r,l}[n]\}_{n=0}^{N-1}$ of the wavelet can be explicitly calculated using the two-scale relations.

Example 12.1 (The first decomposition scale $r = 1$) The SHA spectra are

$$\nu_{1,0}[n] = \sqrt{2}b_0[n] = \sqrt{\frac{u^{2p}[n]}{u_1^{2p}[n]}} \cos^{p}\frac{\pi n}{N},$$

$$\nu_{1,1}[n] = \sqrt{2}\overline{\tilde{b}}_0[n] = \omega^{-n}\sqrt{\frac{u^{2p}[n_r]}{u_1^{p}[n]}} \sin^{p}\frac{\pi n}{N}.$$

Example 12.2 (The second decomposition scale $r = 2$) The SHA spectra are

$$\nu_{2,0}[n] = 2b_0[n]b_1[n], \qquad \nu_{2,1}[n] = 2b_0[n]\overline{\tilde{b}}_1[n],$$

$$\nu_{2,2}[n] = 2\overline{\tilde{b}}_0[n]\overline{\tilde{b}}_1[n], \qquad \nu_{2,3}[n] = 2\overline{\tilde{b}}_0[n]b_1[n].$$

Figure 12.1 displays the wavelet packets from the first and the second decomposition scales with their SHA spectra.

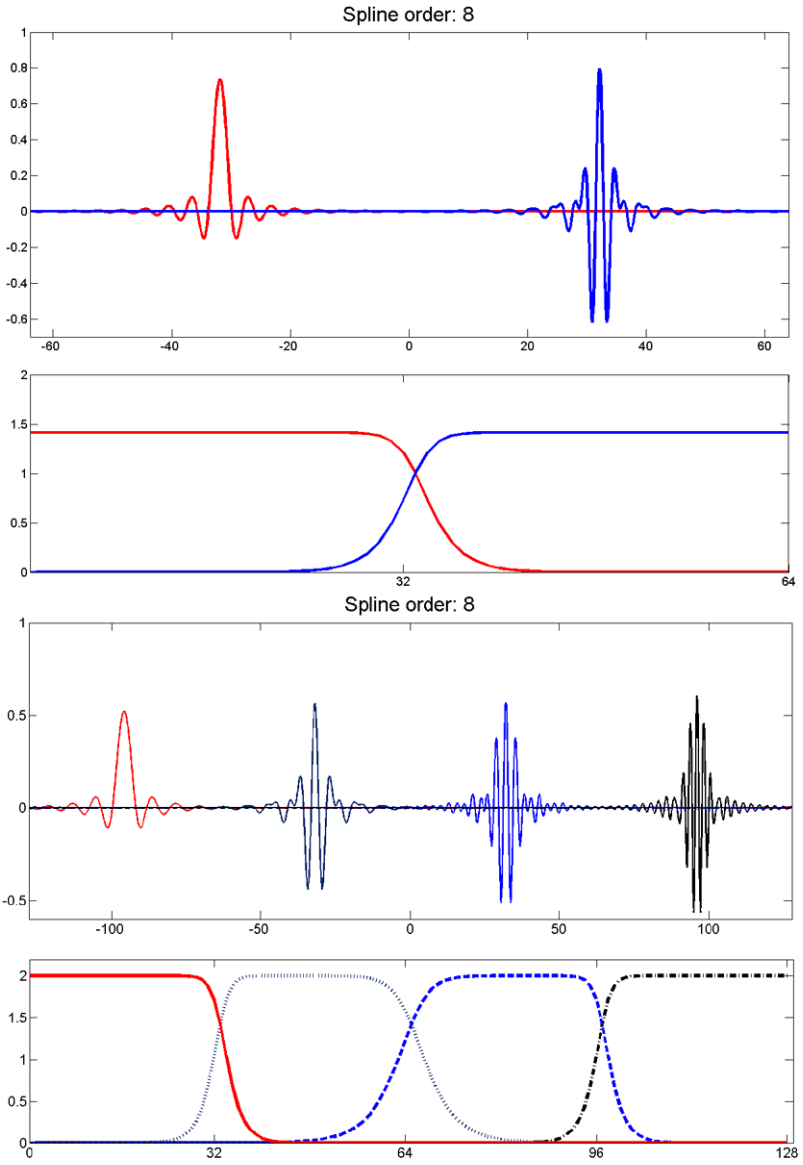


Fig. 12.1 Wavelet packets of order 8 from the 1st (*top*) the 2nd decomposition scales with their SHA spectra (*bottom half-band*)

The wavelet packets are well localized in space. Their SHA spectra have a near rectangular shape (the higher the spline order is the closer the shape is to rectangular) and produce a sequence of partitions of the frequency band. The SHA spectrum

of the wavelet packet $\psi_{r,l}(x)$ is effectively confined within the band

$$\Lambda_{r,l} \stackrel{\text{def}}{=} \left[-\frac{(l+1)N}{2^{r+1}}, -\frac{(l)N}{2^{r+1}}\right] \cup \left[\frac{(l)N}{2^{r+1}}, \frac{(l+1)N}{2^{r+1}}\right], \quad l = 0, \ldots, 2^r - 1. \tag{12.16}$$

Since a spline from ${}^p\mathscr{S}_{r,l}$ is the linear combination of the wavelet packets

$$S_{r,l}(x) = \sum_{k=0}^{N_r-1} q_{r,l}[k]\psi_{r,l}^p\big(x - 2^r k\big), \tag{12.17}$$

then its SHA spectrum is effectively confined within the band $\Lambda_{r,l}$. This provides opportunities to approximate the heat inversion separately in different frequency bands.

12.5 Spline Wavelet Packet Transforms

Let a spline $S(x) \in {}^p\mathscr{S}$ be represented by the orthonormal basis splines

$$S(x) = \sqrt{\frac{1}{N}} \sum_{n=-N/2}^{N/2-1} \sigma[n]\gamma^p[n](x). \tag{12.18}$$

The sequence $\{\sigma[n]\}$, $n = -N/2, \ldots, N/2-1$, is the SHA spectrum of the spline $S(x)$. The space ${}^p\mathscr{S}$ is the orthogonal sum of the subspaces ${}^p\mathscr{S}_{1,0}$ and ${}^p\mathscr{S}_{1,1}$ whose orthonormal bases are $\{\gamma_{1,0}^p[n](x)\}$ and $\{\gamma_{1,1}^p[n](x)\}$, respectively, where $n = 0, \ldots, N_1 - 1$ and $N_1 = N/2$. Thus, $S(x)$ can be represented as the sum of its orthogonal projections onto the subspaces ${}^p\mathscr{S}_{1,i}$, $i = 0, 1$: $S(x) = S_{1,0}(x) \oplus S_{1,1}(x)$, where

$$S_{1,i}(x) \stackrel{\text{def}}{=} \sqrt{\frac{2}{N}} \sum_{n=0}^{N/2-1} \sigma_1[n]\gamma_{1,i}^p[n](x), \quad i = 0, 1. \tag{12.19}$$

The orthonormality of the spline basis implies

$$\sigma[n] = \sqrt{N}\big\langle S, \gamma^p[n]\big\rangle, \qquad \sigma_{1,i}[n] = \sqrt{N/2}\big\langle S, \gamma_{1,i}^p[n]\big\rangle, \quad i = 0, 1. \tag{12.20}$$

By using the two-scale relations given by (12.12) and (12.13), we derive for $n = 0, \ldots, N/2-1$

$$\sigma_{1,0}[n] = \sqrt{N/2}\big\langle S, \gamma_{1,0}^p[n]\big\rangle = \sqrt{\frac{1}{2}}\big(b_0[n]\sigma[n] + b_0[n_1]\sigma[n_1]\big), \tag{12.21}$$

$$\sigma_{1,1}[n] = \sqrt{\frac{1}{2}}\big(\tilde{b}_0[n]\sigma[n] + \tilde{b}_0[n_1]\sigma[n_1]\big), \quad n_1 = n + N/2. \tag{12.22}$$

We can present (12.21) and (12.22) in a matrix form

$$\begin{pmatrix}\sigma_{1,0}[n]\\ \sigma_{1,1}[n]\end{pmatrix}=\sqrt{\frac{1}{2}}\mathbf{A}_0[n]\cdot\begin{pmatrix}\sigma[n]\\ \sigma[n_1]\end{pmatrix},\qquad \mathbf{A}_m[n]\stackrel{\text{def}}{=}\begin{pmatrix}b_m[n]\ b_m[n_{m+1}]\\ \tilde{b}_m[n]\ \tilde{b}_m[n_{m+1}]\end{pmatrix}. \quad (12.23)$$

The coordinates of the projections of $S(x)$ onto the subspaces ${}^p\mathscr{S}_{r,l}$ are derived iteratively:

$$\begin{aligned}\begin{pmatrix}\sigma_{m,2l}[n]\\ \sigma_{m,2l+1}[n]\end{pmatrix}&=\sqrt{\frac{1}{2}}\mathbf{A}_{m-1}[n]\cdot\begin{pmatrix}\sigma_{m-1,l}[n]\\ \sigma_{m-1,1}[n+N_r]\end{pmatrix}\quad \text{if } l \text{ is even},\\ \begin{pmatrix}\sigma_{m,2l+1}[n]\\ \sigma_{m,2l}[n]\end{pmatrix}&=\sqrt{\frac{1}{2}}\mathbf{A}_{m-1}[n]\cdot\begin{pmatrix}\sigma_{m-1,l}[n]\\ \sigma_{m-1,1}[n+N_r]\end{pmatrix}\quad \text{if } l \text{ is odd}.\end{aligned} \quad (12.24)$$

12.6 Wavelet Packet Bases

Assume that the spline $S(x)\in{}^p\mathscr{S}$ is expanded over the orthonormal bases

$$S(x)=\frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}\sigma[n]\gamma^p[n](x)=\sum_{k=0}^{N-1}q[k]\varphi^p(x-k),$$
$$\sigma[n]=\sum_{k=0}^{N-1}\omega^{-nk}q[k]=\hat{q}[n],\quad q[k]=\frac{1}{N}\sum_{n=0}^{N-1}\omega^{nk}\sigma[n].$$

For example, if the samples $S(k)=y[k]$, $k=0,\ldots,N-1$, are available then (12.8) claims that $\sigma[n]=\hat{y}[n]/V[n]$. Iterative application of the transform given by (12.24) expands the projections splines $S_{r,l}(x)\in{}^p\mathscr{S}_{r,l}$ over the orthonormal bases $\{\gamma_{r,l}[n](x)\}$. Then, the coordinates $\{q_{r,l}[k]\}$ of the alternative expansion (12.17) over the orthonormal wavelet packet bases $\{\psi^p_{r,l}(x-2^rk)\}$ are derived by the application of the IDFT: $q_{r,l}[k]=N_r^{-1}\sum_{n=0}^{N_r-1}\omega^{2^rnk}\sigma_{r,l}[n]$. The subspace ${}^p\mathscr{S}_{r-1,l}={}^p\mathscr{S}_{r,2l}\oplus{}^p\mathscr{S}_{r,2l+1}$. The spline $S_{r-1,l}(x)$ can be expanded either over the basis $\{\psi^p_{r-1,l}(x-2^{r-1}k)\}_{k=0}^{N_{r-1}-1}$ or over the combined orthonormal basis $\{\psi^p_{r,2l}(x-2^rk)\}\biguplus\{\psi^p_{r,2l}(x-2^rk)\}_{k=0}^{N_r-1}$. The decision of which basis is preferable is made once a cost function is defined.

Consequently, once the wavelet packet transform of the spline $S(x)$ is implemented, a wide variety of orthonormal wavelet packet bases becomes available. A basis, which is optimal for a given spline with respect to a certain purpose, can be designed by the Best Basis algorithm [4], which compares the cost function of the "parent" spline $S_{r-1,l}(x)$ with the cost of the "offsprings" $S_{r,2l}(x)$ and $S_{r,2l+1}(x)$. Entropy us a typical cost function.

12.7 Parameterized Spline Solution in the Subspace ${}^{p}\mathscr{S}_{r,l}$

12.7.1 Splines from the Subspaces ${}^{p}\mathscr{S}_{r,l}$

The spline $S_{r,l}(x)$, which is the orthogonal projection of the spline $S(x)$ onto the subspace ${}^{p}\mathscr{S}_{r,l}$, is expanded over the orthonormal wavelet packet basis as in (12.17). On the other hand, the spline $S_{r,l}(x)$ belongs to the initial space ${}^{p}\mathscr{S}$ and can be expanded over the orthonormal basis of ${}^{p}\mathscr{S}$

$$S_{r,l}(x) = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} \zeta_{r,l}[n] \gamma^{p}[n](x), \quad \zeta_{r,l}[n] = \nu_{r,l}[n] \hat{q}_{r,l}[n]. \tag{12.25}$$

The coefficients $\nu_{r,l}[n]$ are the SHA spectrum of the wavelet packet $\psi^{p}_{r,l}$. We emphasize that the DFT sequence $\hat{q}_{r,l}[n]$ is N_r-periodic, where $N_r = N/2^r$.

The projection coordinates $\zeta_{r,l}[n]$ can be expressed via the coordinates $\sigma[n]$ of the spline $S(x)$.

Proposition 12.1 *The following representation of the projection coordinates holds*:

$$\zeta_{r,l}[n] = \frac{\nu_{r,l}[n]}{2^r} \sum_{\lambda=-2^r/2}^{2^r/2-1} \sigma[n + \lambda N_r] \overline{\nu}_{r,l}[n + \lambda N_r] \approx \frac{1}{2^r} \sigma[n] \left|\nu_{r,l}[n]\right|^2. \tag{12.26}$$

Remark 12.1 The higher the order p is, the closer $\zeta_{r,l}[n]$ is to $2^{-r}\sigma[n]|\nu_{r,l}[n]|^2$.

Equation (12.9) implies that the application of the operator $\mathbf{V}_t$ to $S_{r,l}(x)$ results in

$$S_{r,l}(x,t) = \mathbf{V}_t S_{r,l}(x) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \eta[n](t) \zeta_{r,l}[n] \gamma^{p}[n](x). \tag{12.27}$$

From (12.6), sampling of the spline $S(x,t) = \mathbf{V}_t S(x)$ becomes

$$y[k] \stackrel{\text{def}}{=} S(k,t) = \frac{1}{N} \sum_{n=0}^{N-1} \eta[n](t) \sigma[n] V[n] \omega^{kn}, \quad V[n] \stackrel{\text{def}}{=} \frac{u^{p}[n]}{\sqrt{u^{2p}[n]}},$$

while sampling of $S_{r,l}(x,t) = \mathbf{V}_t S_{r,l}(x)$

$$y_{r,l}[k] \stackrel{\text{def}}{=} S_{r,l}(k,t) = \frac{1}{N} \sum_{n=0}^{N-1} \eta[n](t) \zeta_{r,l}[n] V[n] \omega^{kn}. \tag{12.28}$$

Equation (12.26) implies the approximated relation

$$\hat{y}_{r,l}[n] \approx \frac{1}{2^r} \eta[n](t) \sigma[n] V[n] \left|\nu_{r,l}[n]\right|^2 = \frac{1}{2^r} \hat{y}[n] \left|\nu_{r,l}[n]\right|^2. \tag{12.29}$$

Remark 12.2 Equation (12.29) can be interpreted in a sense that confinement of the operator's $\mathbf{V}_t$ domain from the whole spline space ${}^p\mathscr{S}$ to the subspace ${}^p\mathscr{S}_{r,l}$ effectively results in multiplication of the DFT $\hat{y}[n]$ of the sampled output $y[k] = \mathbf{V}_t S(k)$ with the factor $2^{-r}|\nu_{r,l}[n]|^2$.

12.7.2 Parameterized Spline Solution

The scheme for a partial solution of Problem 12.1 in the subspace ${}^p\mathscr{S}_{r,l}$ is very similar to the scheme of a global solution presented in Sect. 12.3. By assumption, t is a known time parameter and the vector $\mathbf{z} = \{z[k] = g(k,t) + e_k\} = \mathbf{g} + \mathbf{e}$ is available, where $g(x,t) = \mathbf{U}_t f(x)$. Then, a partial approximated inversion of the heat equation (12.1) is derived as a spline

$$S_{r,l}(x) = \sum_{k=0}^{N_r-1} q_{r,l}[k]\psi_{r,l}^p\left(x - 2^r k\right) = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} \zeta_{r,l}[n]\gamma^p[n](x) \tag{12.30}$$

such that the spline $S_{r,l}(x,t) = \mathbf{V}_t S_{r,l}(x)$ approximates, in some sense, the available discrete data $\mathbf{z}$. To be specific, Remark 12.2 suggests that the sampled spline $S_{r,l}(k,t)$ should approximate the "filtered"

$$\tilde{z}_{r,l}[k] \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=0}^{N-1} \omega^{kn} \hat{\tilde{z}}_{r,l}[n], \quad \text{where } \hat{\tilde{z}}_{r,l}[n] \stackrel{\text{def}}{=} \hat{z}[n] \frac{|\nu_{r,l}[n]|^2}{2^r}, \tag{12.31}$$

rather than the entire data $\mathbf{z}$. Similarly to Sect. 12.3, we find a spline $S(\rho, x) \in {}^p\mathscr{S}_{r,l}$, which minimizes the functional $\rho I(S) + E_{r,l}(S)$, where

$$I(S) \stackrel{\text{def}}{=} \left\|(S)''\right\|^2, \qquad E_{r,l}(S) \stackrel{\text{def}}{=} \sum_k \left(S(k,t) - \tilde{z}_{r,l}[k]\right)^2, \qquad S(x,t) \stackrel{\text{def}}{=} \mathbf{V}_t S(x), \tag{12.32}$$

and ρ is a numerical parameter.

Let a spline $s(x) \in {}^p\mathscr{S}_{r,l}$ be represented as in (12.30)

$$s(x) = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} \zeta_{r,l}[n]\gamma^p[n](x).$$

Then,

$$I(s) = \frac{1}{N} \sum_{n=0}^{N-1} W[n]\left|\zeta_{r,l}[n]\right|^2, \qquad E_{r,l}(s) = \frac{1}{N} \sum_{n=0}^{N-1} \left|\eta[n](t)\zeta_{r,l}[n]V[n] - \hat{\tilde{z}}_{r,l}[n]\right|^2.$$

The sequences $W[n]$ and $V[n]$ are defined in (12.5).

A solution to the minimization problem is the spline from ${}^{p}\mathscr{S}_{r,l}$

$$S_{r,l}(\rho,x)=\sqrt{\frac{1}{N}}\sum_{n=0}^{N-1}\zeta_{r,l}(\rho)[n]\gamma^{p}[n](x),\qquad \zeta_{r,l}(\rho)[n]=\frac{\overline{\eta}[n](t)V[n]\hat{\tilde{z}}_{r,l}(n)}{A[n](\rho)},$$

$$A[n](\rho)\stackrel{\text{def}}{=}\rho W[n]+\left|\eta[n](t)V[n]\right|^{2}.$$

Its samples on the grid points

$$S_{r,l}(\rho,k)=\frac{1}{N}\sum_{n=0}^{N-1}\omega^{kn}\zeta_{r,l}(\rho)[n]V[n]=\frac{1}{N}\sum_{n=0}^{N-1}\omega^{kn}\frac{\overline{\eta}[n](t)V^{2}[n]\hat{\tilde{z}}_{r,l}(n)}{A[n](\rho)}. \quad (12.33)$$

12.7.3 Selection of the Regularization Parameter

Assume that we are able to evaluate the errors vector $\mathbf{e}=\{e_k\}_{k=0}^{N-1}$, $e_k=N^{-1}\times\sum_{n=0}^{N-1}\omega^{kn}\hat{e}[n]$, whose variance $\mathrm{var}(\mathbf{e})=\varepsilon^{2}\approx N^{-1}\sum_{k=0}^{N-1}(e_k)^{2}$. Keeping (12.31) in mind, denote

$$e_{r,l}[k]\stackrel{\text{def}}{=}\frac{1}{N}\sum_{n=0}^{N-1}\omega^{kn}\hat{e}_{r,l}[n],\quad \text{where } \hat{e}_{r,l}[n]\stackrel{\text{def}}{=}\hat{e}[n]\frac{|\nu_{r,l}[n]|^{2}}{2^{r}},$$

$$(\varepsilon_{r,l})^{2}\stackrel{\text{def}}{=}\sum_{k=0}^{N-1}\left(e_{r,l}[k]\right)^{2}=\frac{1}{N}\sum_{n=0}^{N-1}\left|\hat{e}_{r,l}[n]\right|^{2}.$$

The function

$$e_{r,l}(\rho)\stackrel{\text{def}}{=}E_{r,l}\left(S_{r,l}(\rho,\cdot)\right)=\frac{1}{N}\sum_{n=0}^{N-1}\left(\frac{\rho W[n]|\hat{\tilde{z}}_{r,l}[n]|}{A[n](\rho))}\right)^{2}$$

grows monotonically from zero to $N^{-1}\sum_{n=0}^{N-1}|\hat{\tilde{z}}_{r,l}[n]|^{2}=\sum_{k=0}^{N-1}(\tilde{z}_{r,l}[k])^{2}$ as ρ grows from zero to infinity. Therefore, we derive $\rho_{r,l}$ from the equation $e_{r,l}(\rho)=(\varepsilon_{r,l})^{2}$.

12.7.4 Modeling the Noise

We assume that the error vector $\mathbf{e}$ is a zero mean Gaussian white noise. It is seen from (12.2) that the Fourier coefficients of the function $g(x,t)=\mathbf{U}_t f(x)$: $g_n(t)=f_n e^{-t(2\pi n/N)^{2}}$ are fast decaying when n is growing. Thus, the function $g(x,t)$ is efficiently bandlimited. Its significant Fourier coefficients $g_n(t)$ occupy

a relatively narrow band around zero, $-K(t) < n < K(t)$, $K(t) < N/2$, $K(t) \to 0$ as $t \to \infty$. Hence, the DFT coefficients of the data vector $\mathbf{z}$: $\{\hat{z}[n]\} \approx \{\hat{e}[n]\}$ as $n \in [K(t), N/2-1] \cup [-N/2, -K(t)]$. By relying on the fact that the power spectrum $\{|\hat{e}[n]|^2\}$ of the white noise $\mathbf{e}$ is close to a constant for all $n = -N/2, \ldots, N/2-1$, it is possible to evaluate the variance

$$\sigma^2 \approx \frac{1}{(N-K(t))^2} \sum_{n \in [K(t), N/2-1] \cup [-N/2, -K(t)]} \left|\hat{z}[n]\right|^2. \tag{12.34}$$

Then, the noise vector $\mathbf{e}$ is modeled as a zero mean Gaussian random process $\tilde{\mathbf{e}} = \{\tilde{e}_i\}_{i=0}^{N-1}$, whose variance is σ^2. Let $\{\widehat{\tilde{e}}[n]\}_{n=-N/2}^{N/2-1}$ be the DFT spectrum of the model vector $\tilde{\mathbf{e}}$. Then, the values $(\varepsilon_{r,l})^2$, which are needed for the parameter ρ selection, are estimated as

$$(\varepsilon_{r,l})^2 \approx \frac{1}{2^r N} \sum_{n=0}^{N-1} \left|\left(\nu_{r,l}[n]\right)^2 \overline{\widehat{\tilde{e}}[n]}\right|^2. \tag{12.35}$$

Another option for the noise evaluation is to use the scheme in [5].

12.8 Spline Wavelet Packet Solution to Problem 12.1

The partial spline solution $S_{m,l}(\rho, x)$ of the inversion problem in the subspace ${}^p\mathcal{S}_{r,l}$ is derived from the filtered data such that the DFT is $\hat{\tilde{z}}_{m,l}[n] \stackrel{\text{def}}{=} \hat{z}[n]|\nu_{m,l}[n]|^2 2^{-m}$.

To determine an optimal set of the subspaces ${}^p\mathcal{S}_{r,l}$, which reveal the internal structure of the data vector $\mathbf{z}$, we construct the spline $Z(x) = \sum_{n=0}^{N-1} \xi[n] \gamma^p[n](x)$, $\xi[n] = \hat{z}[n]/V[n]$, which interpolates the data $\mathbf{z}$. Then, we apply the Best Basis algorithm to obtain the list $PL = \{(\bar{p}, \bar{l})\}$ such that the shifts of the wavelet packets $\psi_{\bar{m},\bar{l}}$ form an optimal basis for the spline $Z(x)$. The list PL determines the subspaces ${}^p\mathcal{S}^{\bar{m},\bar{l}}$, where the partial solutions for the inversion problem are to be derived. Due to the effective bandlimitedness of the function $g(x,t) = \mathbf{U}_t^1 f(x)$, some subspaces ${}^p\mathcal{S}^{\bar{m},\bar{l}}$, which correspond to higher frequency bands are "empty" in a sense that they, actually, do not contain a contribution from the initial function $f(x)$. Such subspaces are discarded from the list PL.

A scheme for the approximated inversion of the heat equation

1. Calculate the coefficients $\eta[n](t)$ defined in (12.10).
2. Construct the data interpolating spline $Z(x)$.
3. Implement the wavelet packet transform of order p of the spline $Z(x)$.
4. Apply the Best Basis algorithm to the transform coefficients to collect the list PL of relevant subspaces.
5. Reduce the list PL to $\overline{PL}$ by discarding the "empty" subspaces.
6. Evaluate the error vector to estimate the partial variances $(\varepsilon_{p,l})^2$, $(p,l) \in \overline{PL}$, of noise (see (12.35)).

7. Determine the optimal values $\rho_{\bar{m},\bar{l}}$ of the regularization parameter for each pair $(\bar{m},\bar{l}) \in \overline{PL}$.
8. Find the partial solutions $S_{\bar{m},\bar{l}}(\rho_{\bar{m},\bar{l}}, x) \in {}^{p}\mathscr{S}_{\bar{m},\bar{l}}$ for each pair $(\bar{m},\bar{l}) \in \overline{PL}$ (see (12.33)).

The approximated solution to the inversion Problem 12.1 is

$$f(x) \approx S(x) = \sum_{(\bar{m},\bar{l})\in\overline{PL}} S_{\bar{m},\bar{l}}(\rho_{\bar{m},\bar{l}}, x) \in {}^{p}\tilde{\mathscr{S}}.$$

Extension of the algorithm to the 2D case is straightforward once the tensor products of the basis splines are utilized:

$$\gamma^{p}(x,y) \stackrel{\text{def}}{=} \gamma^{p}(x)\gamma^{p}(y), \qquad \varphi^{p}(x,y) \stackrel{\text{def}}{=} \varphi^{p}(x)\varphi^{p}(y),$$

$$\psi^{p}_{r,l,\tilde{l}}(x,y) \stackrel{\text{def}}{=} \psi^{p}_{r,l}(x)\psi^{p}_{r,\tilde{l}}(y).$$

Figure 12.2 displays the SHA spectra of two wavelet packets of order 10 from the second scale. We observe that the spectra have near-parallelepiped shape. The described algorithm can be utilized for signal and image denoising when the time parameter $t = 0$. In this case, the general scheme remains unchanged.

12.9 Numerical Examples

The following are examples, derived from three groups of experiments, on using the block-based methods for 2D images' restoration:

Denoising: Restoration of objects corrupted by Gaussian noise (the time parameter $t = 0$).

Pure blurred input: Restoration of blurred objects when the time parameter $t > 0$ and noise is not known. The advantage of the block based method over the global one materialized in the accurate tuning of the subspaces where the looked for solution is in the effective frequency domain of the blurred image.

Noised blurred input: Restoration of objects from blurred inputs, which were corrupted by Gaussian noise.

These examples illustrate the difference between the performance of the global Tikhonov algorithm (GTA) presented in Sect. 12.3 and of the Best Basis Algorithm (BBA). Visual perception is compared and the peak-signal-to-noise-ratio (PSNR). Three benchmark images each of which is presented by a 512×512 array of samples are used as the initial temperature distributions. The source images for the experiments are shown in Fig. 12.3.

Example 12.3 (Barbara Denoising) The "Barbara" image was corrupted by Gaussian zero-mean noise with standard deviations STD = 25. The time parameter is

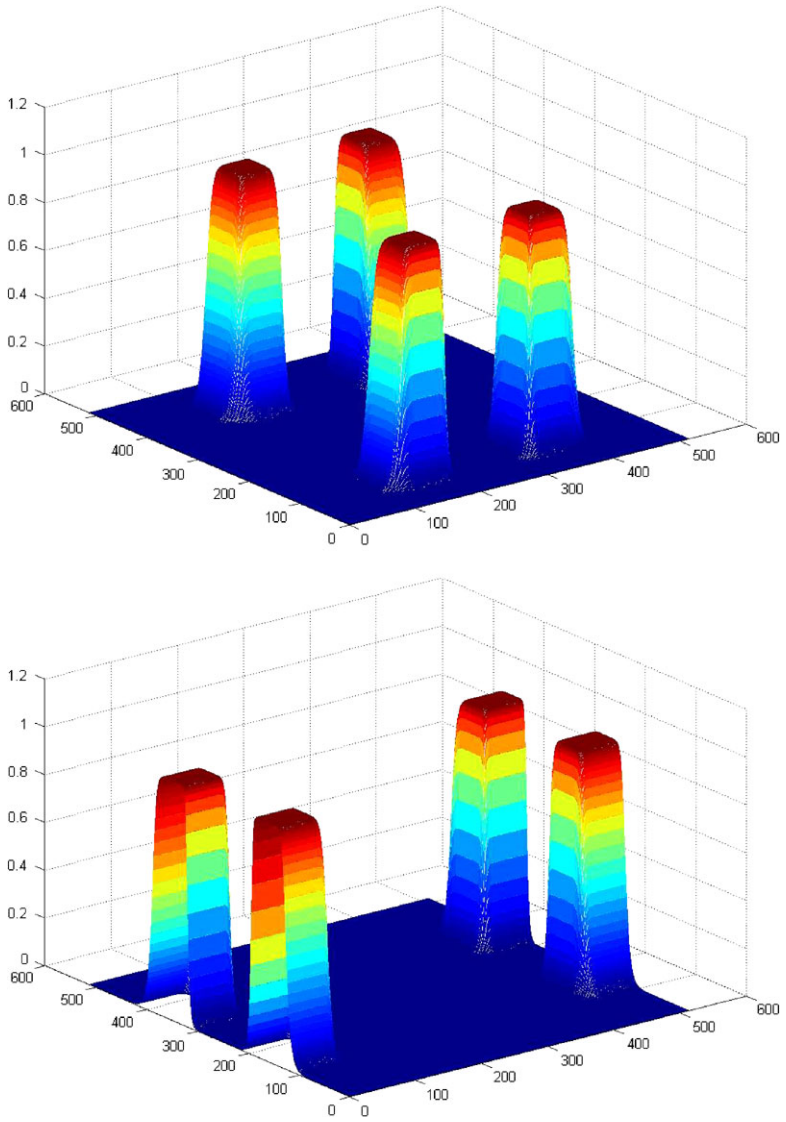


Fig. 12.2 The SHA spectra of wavelet packets of order 10 from the second resolution scale. *Top*: $\psi^{10}_{2,2,3}(x, y)$. *Bottom*: $\psi^{10}_{2,3,1}(x, y)$

$t = 0$. Figure 12.4 displays fragments of the noised input image and of the image that was restored by the applications of GTA and BBA. We observe that BBA produces high PSNR values. The noise was suppressed almost completely. The GTA method did not succeed in noise suppression, although the texture is resolved a little bit better in comparison to BBA.

Fig. 12.3 *Left*: "Barbara". *Center*: "Lena". *Right*: Fingerprint

Fig. 12.4 "Barbara". *Left*: A noised image, STD = 25, PSNR = 20.17. *Center*: An image restored by GTA, PSNR = 24.12. *Right*: An image restored by BBA, PSNR = 25.77, spline wavelet packets of the fourth order from 4 levels were used

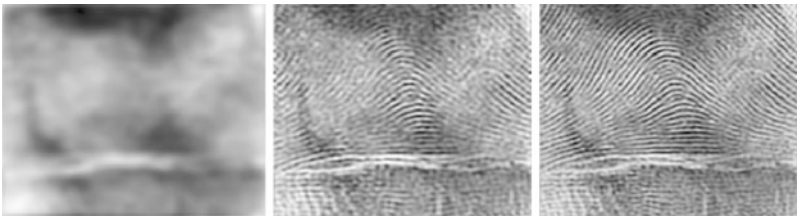


Fig. 12.5 "Fingerprint". *Left*: A blurred image, $t = 46$, PSNR = 15.71. *Center*: An image restored by GTA, PSNR = 17.57. *Right*: An image restored by BBA, PSNR = 20.22, spline wavelet packets of the fourth order from 3 levels were used

Example 12.4 (Restoration of a Strongly Blurred Fingerprint) In this example, the "Fingerprint" image was used as the initial temperature distribution. The input presents the temperature distribution when the time parameter was $t = 46$. The BBA restored the texture of the fingerprint, which was completely smeared in the input. The result produced by GTA was much worse. The results are illustrated in Fig. 12.5.

Example 12.5 (Restoration of Blurred and Noised "Lena") The "Lena" image was used as the initial temperature distribution. The input is the distribution when the time parameter was $t = 2.5$ corrupted by Gaussian noise whose STD = 10. The BBA-restored image is sharper compared to the GTA and its PSNR is higher. See Fig. 12.6 for the results.

Fig. 12.6 "Lena". *Left*: A blurred noised image, $t = 2.5$, noise STD $= 10$, PSNR $= 24.79$. *Center*: An image restored by GTA, PSNR $= 28.22$. *Right*: An image restored by BBA, PSNR $= 28.47$, spline wavelet packets of the fourth order from 4 levels were used

References

1. Averbuch A, Zheludev V (2009) Spline-based deconvolution. Signal Process 89(9):1782–1797
2. Averbuch A, Zheludev V, Khazanovsky M (2011) Deconvolution by matching pursuit using spline wavelet packets dictionaries. Appl Comput Harmon Anal 31(1):98–124
3. Averbuch A, Zheludev V, Neittaanmäki P, Koren J (2010) Block based deconvolution algorithm using spline wavelet packets. J Math Imaging Vis 38(3):197–225
4. Coifman RR, Wickerhauser MV (1992) Entropy-based algorithms for best basis selection. IEEE Trans Inf Theory 38(2):713–718
5. Donoho D, Johnstone I (1994) Ideal spatial adaptation via wavelet shrinkage. Biometrika 81(3):425–455
6. Fourier J (1822) Theorie analytique de la chaleur. Firmin Didot, Paris. Reissued by Cambridge University Press, Cambridge, 2009
7. Tikhonov AN (1963) Solution of incorrectly formulated problems and the regularization method. Sov Math Dokl 4:1035–1038

Chapter 13
Comparison Between Two Multi-Objective Optimization Algorithms: PAES and MGDA. Testing MGDA on Kriging Metamodels

Adrien Zerbinati, Jean-Antoine Désidéri, and Régis Duvigneau

Abstract In multi-objective optimization, the knowledge of the Pareto set provides valuable information on the reachable optimal performance. A number of evolutionary strategies (PAES (Knowles and Corne in Evol. Comput. 8(2):149–172, 2000), NSGA-II (Deb et al. in IEEE Trans. Evol. Comput. 6(2):182–197, 2002), etc.), have been proposed in the literature and proved to be successful in identifying the Pareto set. However, these derivative-free algorithms are very demanding in computational time. Today, in many areas of computational sciences, codes are developed that include the calculation of the gradient, cautiously validated and calibrated. Thus, an alternate method applicable when the gradients are known is introduced presently. Using a clever combination of the gradients, a descent direction common to all criteria is identified. As a natural outcome, the Multiple Gradient Descent Algorithm (MGDA) is defined as a generalization of the steepest descent method and compared with the PAES by numerical experiments. Using the MGDA on a multi-objective optimization problem requires the evaluation of a large number of points with regard to criteria and their gradients. In the particular case of CFD problems, each point evaluation is very costly. Thus here we also propose to construct metamodels and to calculate approximate gradients by local finite differences.

13.1 Introduction

The numerical treatment of a multi-objective minimization is usually aimed at identifying the Pareto set or a convenient subset of it. In the literature, several authors have proposed to achieve this goal by various algorithms, each one adapting a particular Evolution Strategy (ES). Such approaches are compared in the book of Deb

A. Zerbinati (✉) · J.-A. Désidéri · R. Duvigneau
INRIA, 2004 route des lucioles, 06902 Sophia Antipolis, France
e-mail: adrien.zerbinati@inria.fr

J.-A. Désidéri
e-mail: jean-antoine.desideri@inria.fr

R. Duvigneau
e-mail: regis.duvigneau@inria.fr

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_13,

[1]. Using a sufficiently diverse initial sample, these methods produce a discrete set of 2 by 2 non-dominated points. However, the most commonly used methods are very demanding in terms of computational time, as ESs are in general.

In the particular case in which the gradients of the objective functions are at reach, at the current design point, faster algorithms can be developed. In the convex hull of the gradients of the objective functions, a direction exists along which all criteria diminish [2]. The MGDA results in utilizing this direction as search direction and optimizing the step size appropriately. In this way, the classical steepest descent method is generalized to multi-objective optimization. Applying the MGDA thus corresponds to a phase of *cooperative optimization*.

In Sect. 13.2, theoretical aspects leading to the MGDA are briefly recalled. A complete presentation is available in [2]. In Sect. 13.3, the results of a numerical experimentation on a classical test case are presented and commented.

13.2 Theoretical Aspects

13.2.1 Cooperative-Optimization Phase: Multiple-Gradient Descent Algorithm (MGDA)

Here, to be complete, we review briefly the notions developed in [2]. The general context is the simultaneous minimization of n ($n \in \mathbb{N}$) smooth criteria (or disciplines) $J_i(Y)$ (Y is a design vector, $Y \in \mathbb{R}^N$). Starting from an initial design point that is not Pareto optimal, a cooperative optimization phase is defined that is beneficial to all criteria.

13.2.1.1 Pareto Concepts

Following [2], we introduce the notion of *Pareto stationarity*: a design point Y^0 is said to be Pareto stationary if there exists a convex combination of the gradients of the smooth criteria J_i that is equal to 0 at this point. Thus:

Definition 13.1 The smooth criteria $J_i(Y)$ ($1 \leq n \leq N$) are said to be *Pareto stationary* at the design point Y^0 if:

$$\forall i = 1, \ldots, n, \quad u_i^0 = \nabla J_i\left(Y^0\right),$$

$$\exists (\alpha_i)_{i=1,..,n}, \quad \alpha_i \geq 0, \quad \sum_{i=0}^{n} \alpha_i = 1, \quad \sum_{i=0}^{n} \alpha_i u_i^0 = 0.$$

Inversely, if the smooth criteria $J_i(Y)$ ($1 \leq i \leq n$) are not Pareto stationary at the given design point Y^0, a descent direction *common to all criteria* exists.

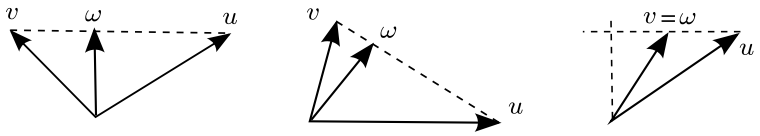


Fig. 13.1 Various possible configurations of the two gradient vectors $u = u_1$ and $v = u_2$ and the minimal norm element ω

13.2.1.2 Existence and Uniqueness of the Minimal Norm Element

Consider a family of vectors, denoted $(u_i)_{i\in I}$, $1 \le i \le n$. The following lemma holds:

Lemma 13.1 (Existence and uniqueness of the minimal norm element) *Assume*

(i) $\{u_i\}$, $1 \le i \le n$, *a family of n vectors in* $\mathbb{R}^N$;
(ii) $\mathcal{U}$ *be the set of strict convex combinations of these vectors*:

$$\mathcal{U} = \left\{ w \in \mathbb{R}^n \;\middle|\; w = \sum_{i=0}^{n} \alpha_i u_i^0;\ \alpha_i > 0,\ \forall i;\ \sum_{i=0}^{n} \alpha_i = 1 \right\}$$

and $\overline{\mathcal{U}}$ *its closure, or convex hull.*

Then,

$$\exists! \omega \in \overline{\mathcal{U}},\ \forall \bar{u} \in \overline{\mathcal{U}} : (\bar{u}, \omega) \ge (\omega, \omega) = \|\omega\|^2.$$

(The element ω exists since $\overline{\mathcal{U}}$ is closed, and it is unique since $\overline{\mathcal{U}}$ is convex; as a result, for all $\bar{u} \in \overline{\mathcal{U}}$, and for all $\varepsilon \in [0, 1]$, $\omega + \varepsilon(\bar{u} - \omega) \in \overline{\mathcal{U}}$, and $\|\omega + \varepsilon(\bar{u} - \omega)\| \ge \|\omega\|$, and this yields the conclusion [2].)

In the case of two criteria, three configurations of the two gradients can be considered, as illustrated in Fig. 13.1. This result applies, in particular, to u_i for all i. But, (u_i, ω) is the Frechet derivative of J_i in the direction ω. Hence, if $\omega \neq 0$, the Frechet derivatives of all the criteria are bounded from below by the strictly positive number $\|\omega\|^2$. The direction $-\omega$ is therefore a descent direction common to all criteria. These considerations yield the following:

Theorem 13.1 *Let* $J_i(Y)$, $1 \le i \le n \le N$, $N \in \mathbb{N}$, *be n smooth functions of the vector* $Y \in \mathbb{R}^N$. *Assume* Y^0 *is an admissible design point. We denote* $u_i^0 = \nabla J_i(Y^0)$ *and*

$$\mathcal{U} = \left\{ w \in \mathbb{R}^N \;\middle|\; w = \sum_{i=1}^{n} \alpha_i u_i^0;\ \forall i,\ \alpha_i > 0;\ \sum_{i=1}^{n} \alpha_i = 1 \right\}. \tag{13.1}$$

Let ω *be the minimal norm element of the convex hull* $\overline{\mathcal{U}}$, *closure of* $\mathcal{U}$. *Then*

(i) *Either* $\omega = 0$, *and the criteria* $J_i(Y)$, $1 \le i \le n$, *are Pareto stationary*;

(ii) *Or $\omega \neq 0$ and $-\omega$ is a descent direction common to all the criteria*; *additionally, if $\omega \in \mathscr{U}$, the inner product $(\bar{u}, \omega)$ is equal to $\|\omega\|^2$ for all $\bar{u} \in \overline{\mathscr{U}}$.*

Based on these results, when the gradients of all the criteria can be computed, the following algorithm (MGDA) proceeds by successive steps that are beneficial to all the criteria. In the practical implementation, one specifies a tolerance ε_{TOL} on $\|\omega\|$ below which the line search is not performed.

13.2.2 Convergence of the MGDA

Provided that the criteria are formulated to be smooth, positive and infinite at infinity, the sequence of iterates produced by the MGDA has been proved to admit a subsequence converging to a Pareto optimal point [2]. One main purpose of this report is to illustrate this convergence by numerical experiments using test cases of variable complexity.

13.2.3 Practical Determination of the Vector ω

In the general case ($n > 2$), ω can be calculated by numerical minimization of the quadratic form that expresses $\|\omega\|^2$ in terms of the coefficients $\{\alpha_i\}$ of the convex combination, subject to the inequality constraints $\alpha_i \geq 0$, for all i, and the linear equality constraint $\sum_i \alpha_i = 1$. Many routines are effective in performing this optimization, for instance, certain evolution strategies. However, the problem may become ill-conditioned for large dimensions.

In the particular case of two objectives, ω can be expressed explicitly. Recall Fig. 13.1, for which $u = u_1 = \nabla J_1$ and $v = u_2 = \nabla J_2$. In this figure, the gradient vectors, elements of $\mathbb{R}^N$ are represented as vectors of $\mathbb{R}^2$ with the same origin O. This results in no loss of generality since only the norms of the two vectors and the angle between them do matter. Eliminating the trivial case in which $u = v$ (for which $\omega = u = v$), the convex hull is then represented by the segment uv connecting the extremities of these representative vectors. Let $\omega^\perp$ be the vector whose origin is O, and the extremity is the orthogonal projection of O onto the line that supports the segment uv (convex hull). If the vector $\omega^\perp$ is in the convex hull, that is, if its representative points are situated on the segment uv, it is ω; otherwise, ω is the vector of the smallest norm between u and v. Thus let

$$\omega = (1 - \alpha)u + \alpha v \tag{13.2}$$

and compute $\alpha^\perp$ for which the above convex combination is orthogonal to $u - v$, that is,

$$\alpha^\perp = \frac{(u, u - v)}{(u - v, u - v)}.$$

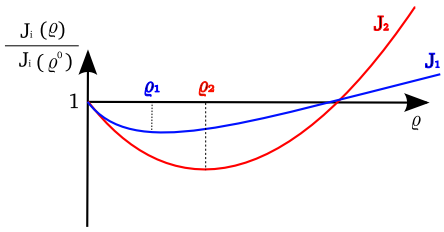


Fig. 13.2 Variation of the *normalized* cost functions with the step size ρ in the $-\omega$ direction

If $\alpha^{\perp} \in [0, 1]$, $\alpha = \alpha^{\perp}$; otherwise, $\alpha = 0$ or 1, that is, $\omega = u$ or v, depending on whether $\alpha^{\perp} < 0$ or > 1.

13.2.4 Line Search

This part deals with the determination of the step length (line search). In multi-criteria optimization, it is not easy to compute a satisfactory step with respect to all the criteria producing a significant evolution. An adaptative method to compute a satisfactory step for each multi-objective problem would be convenient.

At the current design point, the Frechet derivatives of all the criteria are strictly negative (and equal if $\omega \in \mathscr{U}$). For each criterion, a surrogate quadratic model is constructed after computing three function values, and a related optimum step size ρ_i is calculated corresponding to the location of the ith surrogate model's minimum (see Fig. 13.2).

Now, we choose the global step ρ as the smallest ρ_i

$$\rho = \min_{i, 1 \le i \le n} \rho_i .$$

The vector ω is such that, for all i, $\rho_i \ge 0$ and $\rho \ge 0$. Whenever $\rho = 0$, the MGDA is interrupted.

13.3 Numerical Experimentation

In this section, we conduct numerical experiments to demonstrate the convergence of the MGDA to Pareto optimal solutions, and to compare this algorithm with the PAES [4].

13.3.1 Fonseca Test Case

This test case corresponds to the two-objective unconstrained minimization of the functions

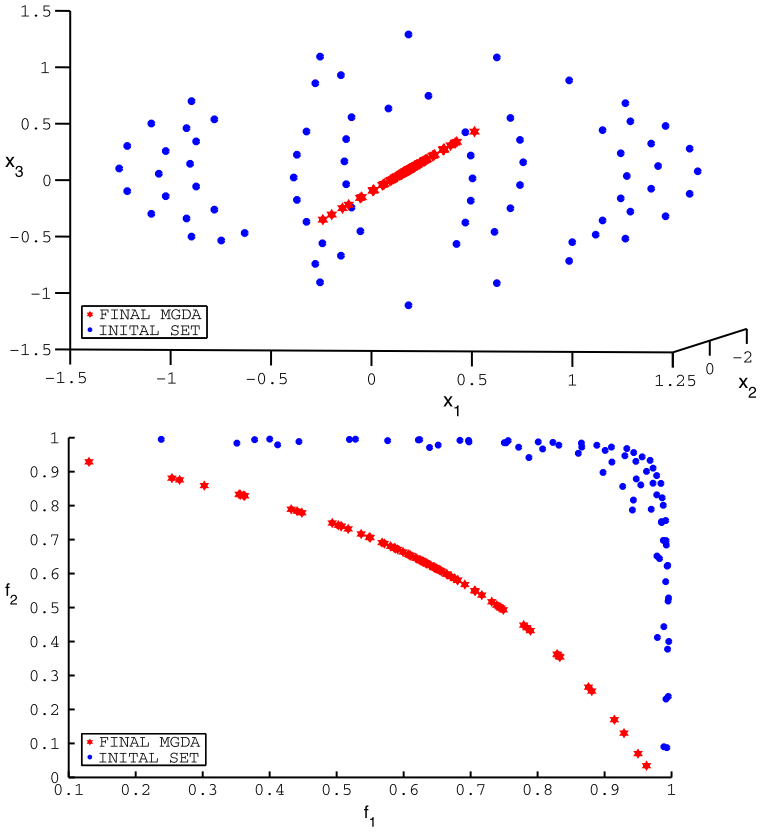


Fig. 13.3 Convergence of the MGDA to the Pareto front, for several initial design points in the design space (x, y, z) (*top*) and in the function space (f_1, f_2) (*bottom*)

$$\begin{cases} f_1(x) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)^2\right), \\ f_2(x) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i + \frac{1}{\sqrt{3}}\right)^2\right). \end{cases}$$

The design variable is $x = (x_1, x_2, x_3) \in \mathbb{R}^3$. This test case is known to yield a continuous but concave Pareto set in the function space. Here, the Pareto set is not known analytically, but has been well identified by Deb using the well-known genetic algorithm NSGA-II [1]. To obtain an accurate discrete representation of the Pareto set by the MGDA, we have applied the method starting from a set of some 50 initial design points located on a sphere in the design-space (Fig. 13.3). In all

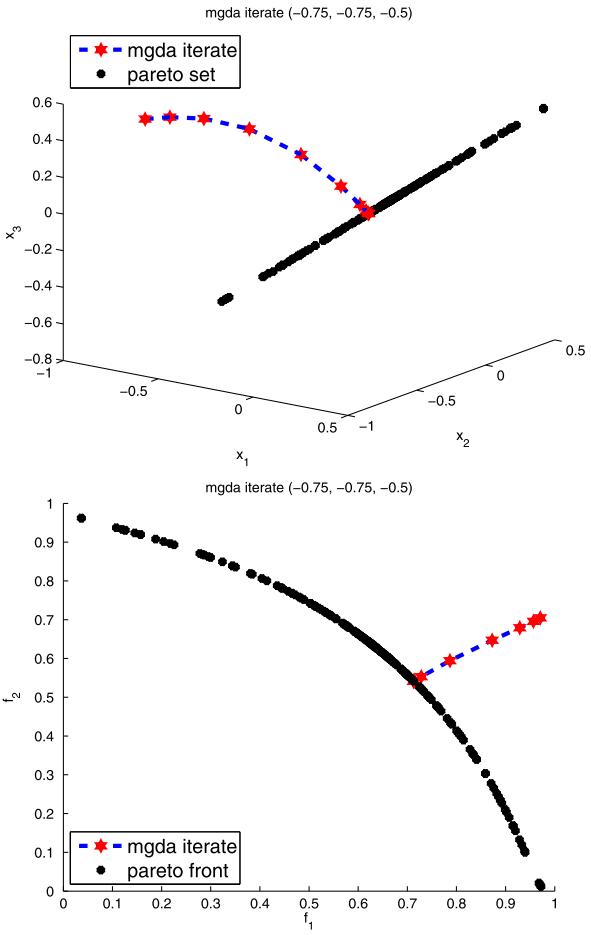


Fig. 13.4 (The Fonseca test case) Convergence of the MGDA to the Pareto front, for several initial design points

cases, the MGDA converges and provides an accurately defined point on the Pareto set (Fig. 13.4).

In the next experiment, we have first applied the PAES twice, each time starting from a different design point and generating 50 others. Then the remaining dominated design points have been discarded. Thus less than one hundred design points have been archived. This set is compared in Fig. 13.5 with the result of applying the MGDA starting from 12 well-distributed initial design points, so that the number of function evaluations is the same in the two cases. The MGDA again produces design points closer to the Pareto set (improved accuracy), but here in fewer numbers.

However, at an identical computational cost, generally, the PAES introduces more diversity in the final result. Thus it appears interesting to combine the accuracy of the MGDA with the robustness of the PAES in a hybrid method. To check

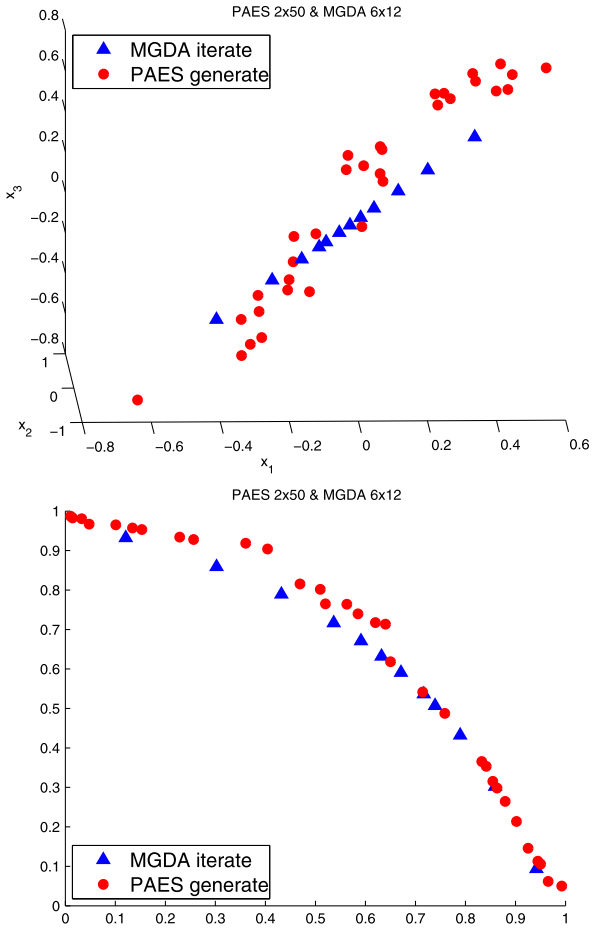


Fig. 13.5 (The Fonseca test case) A Pareto set approximated discretely by the PAES and the MGDA

this, we have used the two methods sequentially: the PAES first to generate 15 design points, retaining eight nondominated design points, then used as initial points for MGDA. In each case about three to four iterations are sufficient to converge and produce the accurate result indicated in Fig. 13.6.

13.4 Applying MGDA on a Kriging Metamodel

In this section, we conduct numerical experiments to demonstrate the convergence of the MGDA to Pareto optimal solutions in conjunction with Kriging metamodels. The first Kriging metamodel is constructed with an initial database. From each initial point, the MGDA yields a better point used subsequently to update the metamodel.

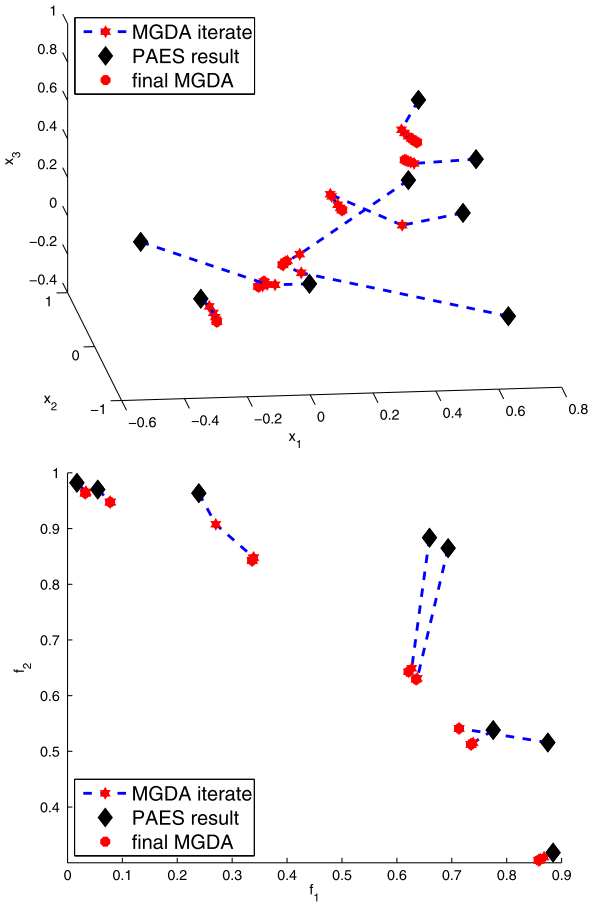


Fig. 13.6 (The Fonseca test case) The first step with a large PAES followed by the MGDA iterates on each non-dominated point found. The design space (*top*) and the functional space (*bottom*)

13.4.1 Kur Test Case

This test case corresponds to the two-objective unconstrained minimization of the functions

$$g_1(x) = -\sum_{i=1}^{2} -10\exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right), \qquad g_2(x) = \sum_{i=1}^{3}\left(|x_i|^{0.8} + 0.5\sin\left(x_i^3\right)\right).$$

The design variable is $x = (x_1, x_2, x_3) \in \mathbb{R}^3$. This testcase is known to yield a non-convex discontinuous Pareto set in the function space. Two generations of non dominated points applying the PAES from different initial configurations gives a good discrete approximation of the Pareto front obtained by Deb [1]. Figure 13.7 shows that the Pareto set here is discontinuous, especially in the design space, where three distinct groups of points are evident.

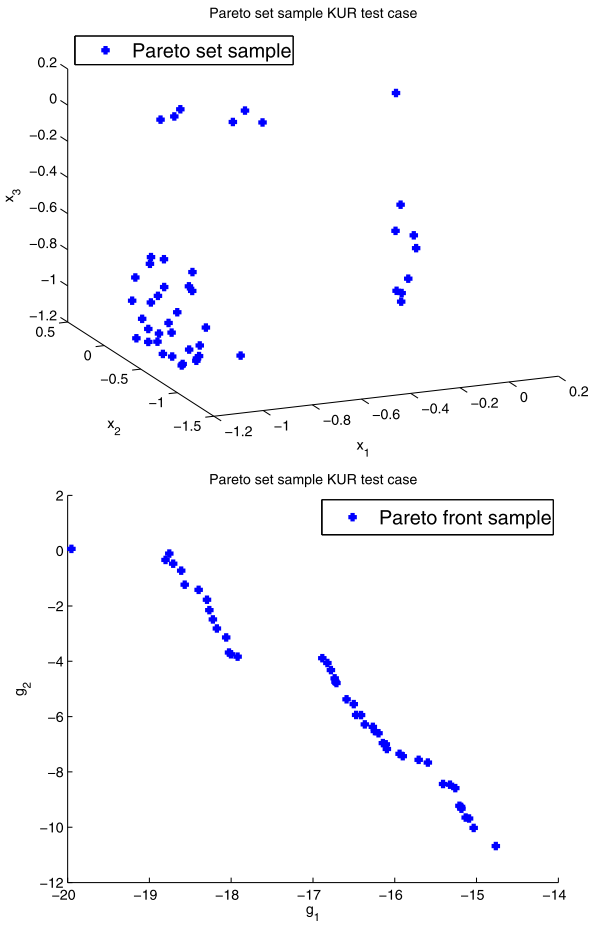


Fig. 13.7 (The Kur test case) A discrete Pareto front produced by two generations of PAES optimization

In the next experiment, we have first applied the PAES once from one initial design point to generate 100 new points. The PAES sorts out 11 non-dominated points from these 100. For each point obtained, the MGDA produces a new one closer to the Pareto front, as illustrated by Fig. 13.8.

Because of the sine in the second function, this test case is a multi modal problem. Thus optimization algorithms based on gradient descent methods have experienced difficulties. To assess the MGDA, a clever strategy must be adopted to generate a sufficiently diverse set of initial points. Presently we use an initial small and diverse set of design points forming a sample of a latin hypercube. This set gives a Kriging metamodel on which the MGDA drives each initial point to a better one in terms of function values. If the MGDA points are sufficiently widespread, a new metamodel is constructed with the initial set augmented. Whenever a new point is found too close to another one in the database, it is not considered to update the metamodel. In a few iterations of this method, the best points obtained are close to the Pareto front.

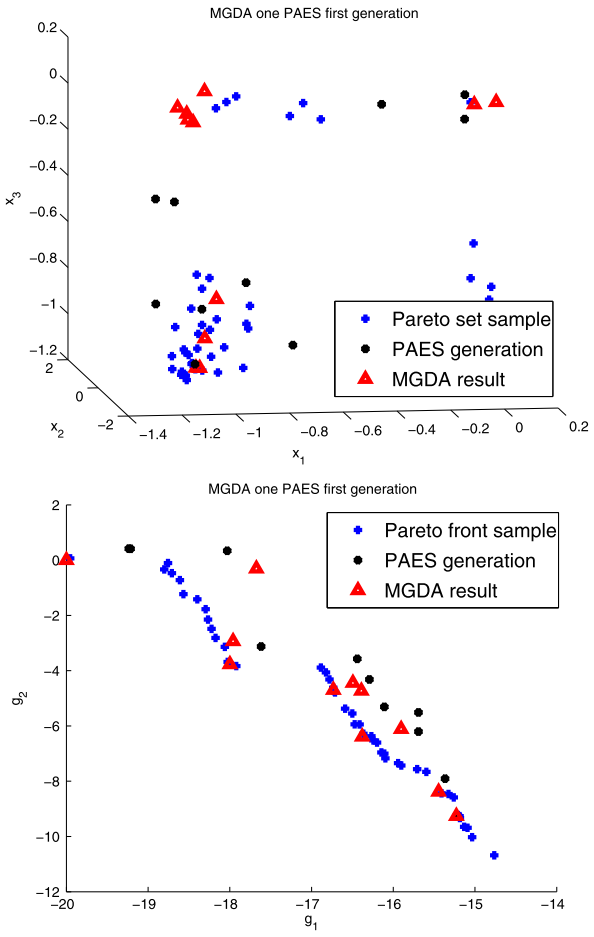


Fig. 13.8 (The Kur test case) Applying the MGDA to each non-dominated point from one PAES generation of 100 points (sort 11 non-dominated)

The following experiment (Fig. 13.9) is based on a set of 10 initial design points from $[-5, 5]^3$ evaluated with respect to g_1 and g_2, after 10 iterations of the process described above. The computational cost corresponds to 43 point evaluations, including the database but not the metamodel construction.

13.4.2 CFD Test Case

The last experiment is an optimum-shape design in compressible aerodynamics. The transonic flow ($M_\infty = 0.83$, $\alpha = AoA = 2^o$) about a generic aircraft wing is simulated by the solution of the 3D Euler equations by an upwind finite volume method over an unstructured mesh of some 200,000 points generated by the software GMSH [3]. The cross sections of the wing are made homothetic with a linear variation in the spanwise direction. Thus only the shape of these sections, an airfoil, is optimized.

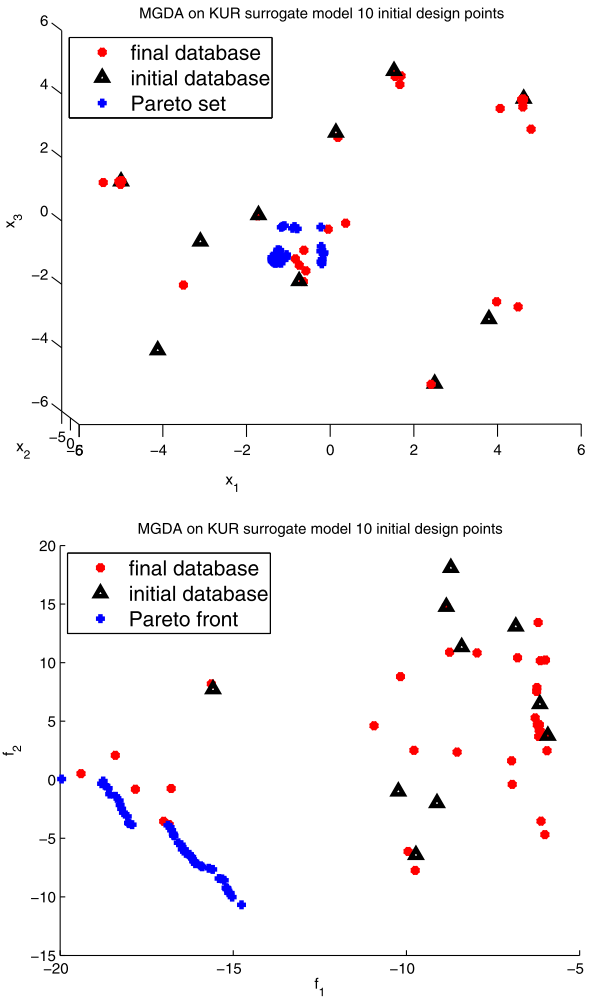


Fig. 13.9 (The Kur test case) Evolution of points given by the MGDA on an evoluting Kriging metamodel. 10 initial design points lead to 43 points

This airfoil is represented by seven B-spline functions for the upper surface, and seven other ones for the lower surface. The leading and trailing edges are fixed, and this permits us to introduce a total of 10 geometrical design variables. Initially, these variables are set to define a cross section close to the classical NACA0012 airfoil.

The MGDA is used here to solve the two-criteria optimization problem consisting of maximizing the lift coefficient and minimizing the drag coefficient simultaneously, starting from the specified initial geometry.

An initial set of 40 design points forming a sample of a latin hypercube in $\mathbb{R}^{10}$ has been considered. This first set of data points is employed for two purposes. Firstly, it is used to construct initial Kriging metamodels of both functions (lift and drag). Secondly, it is used throughout the following cycle to provide starting points to initiate the MGDA iteration in different conditions. This iteration is conducted until conver-

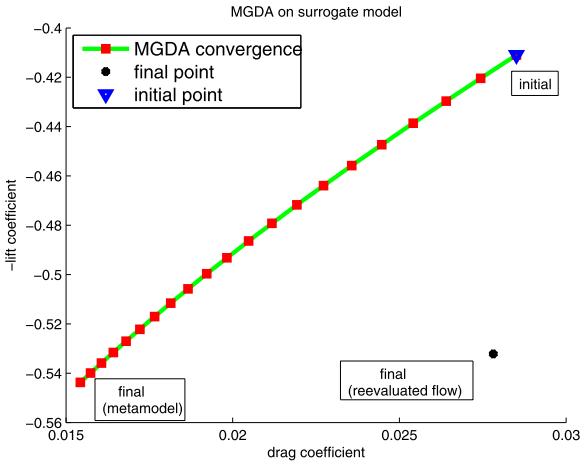


Fig. 13.10 (The Eulerian flow test case) An example of the convergence of the MGDA from an initial database point on the metamodel and the corresponding simulation result point

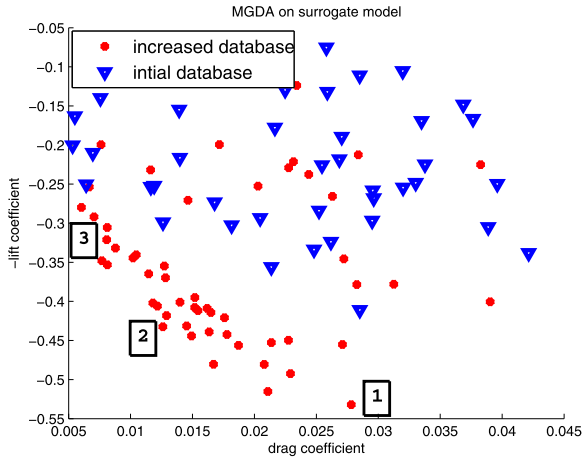


Fig. 13.11 (The Eulerian flow test case) Evolution of data points by the MGDA applied to Kriging metamodels of lift and drag; the dataset is made of 40 design points initially, and 95 ultimately. All points are the result of Eulerian simulation

gence using at every iteration gradients that are calculated by local finite-differences of the metamodels. Each converged point belongs to the Pareto set associated with the two-criteria problem related to the metamodels. It is then re-evaluated by a flow computation and added to the database unless it is found too close to an existing point. At the completion of this database enrichment process, the metamodels are updated, and this completes the cycle. In practice, in what follows, only two cycles were performed.

Figure 13.10 represents the convergence of the MGDA from a particular initial database point. The figure indicates the converged point and the point obtained by the same design re-evaluated by an Euler flow computation (actual lift and drag).

Figure 13.11 represents the initial database of 40 points and the ultimate database. With only 95 calls to the CFD solver, a significant improvement of both criteria is achieved and, visibly, an approximate Pareto front begins to form.

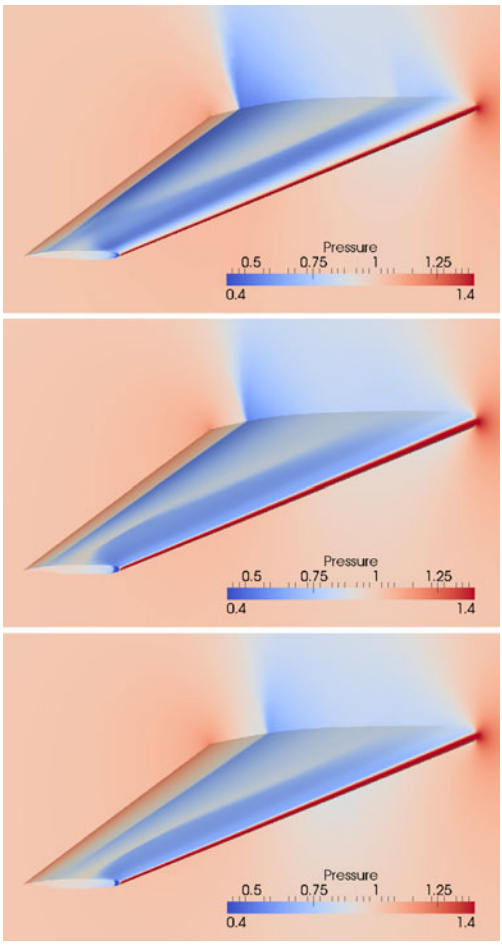


Fig. 13.12 A pressure field associated with design points 1, 2, and 3 of Fig. 13.11

Figure 13.12 represents pressure fields on the wing and the symmetry plane, corresponding to three particular non-dominated points of the ultimate database (points 1 and 3 of Fig. 13.11 on top and bottom respectively). Point 1 corresponds to the flow with the shock wave of the strongest intensity of the three; it produces the largest values of both lift and drag. Inversely, point 3 is associated with the smallest values, and point 2 with intermediate.

13.5 Conclusion

In this article, we have tested by numerical experiment a recently proposed gradient-based algorithm for multi-objective optimization, the MGDA [2].

Firstly, the convergence to Pareto optimal solutions has been demonstrated in an analytical test case corresponding to a continuous but concave Pareto front (the Fonseca test case). Additional information on this comparison can be found in [5].

Secondly, the MGDA has been compared with the well-known PAES algorithm. Both the Fonseca and Kur test cases have been considered in this comparison. We found that the two algorithms have particular merits of their own. The PAES is very effective in converging to a very diverse dataset, whereas the MGDA achieves this only if the initial set of design points is itself diverse. However, the iterative convergence of the MGDA which makes use of (approximate) gradients is much faster. Thus both algorithms are complementary.

Thirdly, a hybrid method has been proposed and tested over the above mathematical test cases, demonstrating promising potentials.

Lastly, in the context of a two-objective aerodynamic wing shape optimization in which the 3D Euler equations have been solved, the MGDA has been used to define a strategy to progressively enrich the database associated with metamodels of drag and lift. With less than 100 calls to the flow solver, both lift and drag have been improved significantly from an initial design of a wing whose cross section was close to the classical NACA0012 airfoil.

References

1. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
2. Désidéri J-A (2009) Multiple-gradient descent algorithm (MGDA). INRIA research report 6953, June. http://hal.inria.fr/inria-00389811/en/
3. Geuzaine C, Remacle JF (2010) A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. Version 2.5.0, October. http://geuz.org/gmsh/
4. Knowles JD, Corne DW (2000) Approximating the nondominated front using the Pareto Archived Evolution Strategy. Evol Comput 8(2):149–172
5. Zerbinati A, Désidéri J-A, Duvigneau R (2011) Comparison between MGDA and PAES for multi objective optimization. INRIA research report 7667, June. http://hal.inria.fr/docs/00/60/54/23/PDF/RR-7667.pdf

Chapter 14
Polar Classification of Nominal Data

Guy Wolf, Shachar Harussi, Yaniv Shmueli, and Amir Averbuch

Abstract Many modern systems record various types of parameter values. Numerical values are relatively convenient for data analysis tools because there are many methods to measure distances and similarities between them. The application of dimensionality reduction techniques for data sets with such values is also a well known practice. Nominal (i.e., categorical) values, on the other hand, encompass some problems for current methods. Most of all, there is no meaningful distance between possible nominal values, which are either equal or unequal to each other. Since many dimensionality reduction methods rely on preserving some form of similarity or distance measure, their application to such data sets is not straightforward. We propose a method to achieve clustering of such data sets by applying the diffusion maps methodology to it. Our method is based on a distance metric that utilizes the effect of the boolean nature of similarities between nominal values (i.e., equal or unequal) on the diffusion kernel and, in turn, on the embedded space resulting from its principal components. We use a multi-view approach by analyzing small, closely related, sets of parameters at a time instead of the whole data set. This way, we achieve a comprehensive understanding of the data set from many points of view.

Keywords Clustering · Unsupervised learning · Diffusion maps · Nominal data

G. Wolf (✉) · S. Harussi · Y. Shmueli · A. Averbuch
School of Computer Science, Tel Aviv University, P.O. Box 39040, Tel Aviv 69978, Israel
e-mail: guy.wolf@cs.tau.ac.il

S. Harussi
e-mail: harussis@tau.ac.il

Y. Shmueli
e-mail: yaniv.shmueli@gmail.com

A. Averbuch
e-mail: amir@math.tau.ac.il

G. Wolf · S. Harussi · A. Averbuch
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_14,

14.1 Introduction

One of the most sought after tasks nowadays is that of finding patterns and structures in large volumes of high dimensional data. As storage becomes cheaper, network bandwidth increases and sampling technologies become more advanced, the amounts of data collected from various systems increase exponentially. A common trend in many applications is to log and record every action of the system for future analysis. In particular, errors and exceptions are common types of massively recorded items.

The task of unsupervised learning of high-dimensional data has been studied extensively in statistical and machine learning literature. Usually, an assumption concerning the underlying structure of the data is made. One common assumption is that the data consist of classes that represent some form of similarity between data points from the same class. Detecting the classes and classifying the data points is often done by clustering algorithms applied to a data representation, which preserves some desired properties (i.e., similarities) of the data set.

Classical clustering algorithms are loosely divided into two major categories. *Partitional* algorithms aim at finding an optimal partition of the data set into the desired clusters. *Hierarchical* algorithms, on the other hand, aim at constructing a hierarchy of clusters from the data. This is usually done in several iterations, each refining the previous one while providing the hierarchy with an additional level. Classic partitioned algorithms are k-Means [27] and its variants (e.g., Fuzzy c-Means [3] and k-Prototypes [22], which adapts k-Means to handle categorical values). Typical hierarchical algorithms are BIRCH [42], CURE [18], and Chameleon [25]. Modern algorithms also use additional approaches. Density-based clustering algorithms, such as DBSCAN [16], DENCLUE [20], and OPTICS [2], define clusters as dense areas separated by sparse ones. Grid-based clustering algorithms analyze cells rather than single samples, thus being more efficient computationally. Some typical examples of such algorithms are STING [37], STING+ [38], WaveCluster [32], CLIQUE [1], GDILC [43], and Localized Diffusion Folders [12].

The data types handled by a clustering algorithm can be divided into three major categories [36]: numerical, nominal, and transactional. Most of the study of clustering algorithms deals with numerical data sets, in which there is a relatively simple notion of proximity or similarity between samples. Most of the algorithms mentioned above deal with numerical data; additional examples can be found in [4, 17, 24].

While there are significantly less clustering algorithms designed for handling nominal data, some classic examples do exist. Notable examples of such clustering algorithms are k-Modes [21] and ROCK [19], both of which deal directly with nominal data, and OPOSSUM [34], which deals with ordinal data (i.e., discrete values with order). A different approach is to transform the categories in the data to numerical values. This can be done either by using some order between them or by using binary encoding (1 means a category that appeared for a sample while 0 means the opposite), which would result in a large but very sparse data set. Some examples of this approach can be found in [31, 33].

The last category (i.e., transactional data) is poorly structured in the sense that each sample, also called transaction, contains a variable set of values describing it. Since there is no constant order to the properties of an entry, and their amount may change from sample to sample, comparing samples in such data sets becomes a fairly complicated task. Such data sets may, in some cases, be flattened and reformatted into a nominal format (e.g., by setting an absolute order to the properties and using a special value to express N/A values), in which case the previously mentioned algorithms can be applied. Some examples of algorithms that directly analyze transactional data are LargeItem [35], SLR [41] and CLOPE [39]. Some recent methods for analyzing both nominal and transactional data can be found in [36].

In recent years, dimensionality reduction techniques were used to obtain low-dimensional representations that amplify the similarities between data points. A popular and successful dimensionality reduction method for this purpose is Diffusion Maps (DM) [8, 26]. This method is based on defining the similarities between data points by using a diffusion kernel, which describes a diffusion process (i.e., random walks) on the data set. The first few eigenvectors of this kernel can be used to obtain a low-dimensional representation of the data set, in which the Euclidean distances between data points correspond to random-walk distances, also called diffusion distances, between their original (high-dimensional) counterparts.

Usually, classes of similar data points appear, in the resulting low-dimensional space, as dense clusters separated by sparse areas [10]. By using a density function one can detect these clusters and the data points within them and thus achieve the desired analysis. This methodology was applied for classification and anomaly detection tasks [10]. In the case of anomaly detection, the classes were considered to represent normal behavior while data points that did not belong to any class were considered to be anomalous.

The DM methodology is based on similarities defined by a suitable distance metric. A Gaussian kernel is then used to give the notion of neighborhoods and, with proper normalization over each neighborhood, the diffusion kernel is obtained. When the data contains numeric measurements, there is a wide variety of distance metrics that can be used. Common metrics are the l_1 and l_2 metrics, which give good results in many practices. When, on the other hand, the data contains nominal values, finding a suitable distance metric is less obvious, as nominal values can be either equal or unequal with no notion of distance or proximity between different values.

One recent approach for handling nominal-valued data sets uses the Hamming distance as the metric that the diffusion kernel is based on. A method, which is based on it, to analyze mixed data sets containing both numeric- and nominal-valued parameters is presented in [13]. This approach can prove useful when there is a one-to-one correspondence between the rows in the data set and the analyzed items, and no bias is created by the dependencies between the parameters of the data set.

In this paper, we deal with a more general nominal-valued data set. We allow several data rows to be related to the same analyzed item. Also, we do not assume that the parameters of the data set are unrelated and we take into account possible bias due to dependencies between them. We define two possible distance metrics

that can be seen as an extension of the Hamming distance. We apply the DM method using these metrics to analyze items, each of which is represented by several rows from the data set.

The structure of the embedded space, which is achieved by the described method, is uniquely different from the ones that appear in many other studies. Instead of similar items being concentrated in dense clusters, these items form rays emanating from a common center near the origin. This unique geometry is a result of the discrete nature of the used distance metric and the resulting diffusion kernel. The clusters are thus identified as having common directions. The rays are not dense when represented by Cartesian coordinates. In polar coordinates, or at least those that correspond to angles, the data points on the same ray are very similar. This observation provides a clustering method to be applied to the embedded space and the method thus called polar clustering.

In addition to the new distance metric used in this paper, we also use a multi-view approach for analyzing the data in an unbiased manner. Multi-view techniques have been applied to many data analysis problems. In these problems, the studied samples consist of different subsets of parameters that, in some cases, even come from different sources. Each of these subsets contributes partial knowledge for the clustering process. Fusing them together can lead to an improved solution. This is done by utilizing the agreements among different views, each representing a single subset. The challenge in these techniques is to find the right parameter partition into subsets, and to understand the weight of each subset and its potential contribution to the learning process. Then, one needs to apply proper normalization and blending methods between the subsets while overcoming problems like cross-dependencies, normalization, repetition, and over- or under-weighting of parameters and subsets.

One common method for dealing with multiple sources (or subsets) of data parameters is to simply ignore the distinctions and concatenate parameters from all the sources into one vector. This represents an implicit assumption that all the parameters, from all the sources, are directly comparable, which is usually not true. Multi-view methods, on the other hand, consider the differences between the subsets and use them to better train the classifier that will be used to analyze the data. One method for applying such a technique is to design a special graph that is based on multiple sources and to use the kernel induced by the graph as the input for a kernel based clustering algorithm [15].

Other multi-view algorithms train two independent classifiers that bootstrap by providing each other with labels for the unlabeled data. The training algorithms tend to maximize the agreement between the two independent classifiers [6, 40]. It has also been shown that the disagreement between two independent classifiers is an upper bound for the error rate of a classifier achieved by uniting them together [9]. This could explain the recent success of multi-view learning in motivating clustering methods that are based on a multi-view approach.

Multi-view classification methods are sometimes called, in the literature, co-training or co-clustering. Under these names, they have been studied thoroughly in [5], where multi-view versions were presented for familiar partitioning methods, such as k-Means, k-Medoids, and EM. Another method that can be used is to

construct a specific kernel (similarity) matrix for each view, and then to blend the matrices into a single kernel matrix. This combined kernel can be used to apply further analysis to the data as a whole, e.g., by training a support vector machine that is based on it [14].

The application of multi-view approaches in conjunction with the DM methodology can be seen in [10, 12, 28]. These works use a hierarchy of views to provide a complete analysis of the data. Construction of each level in the hierarchy by pruning clusters in the previous level and determining the affinities between the pruned clusters is given in [10, 12] whose theoretical justification is given in [11]. This affinity is based on the relations revealed by examining small views, each of which contains samples in the two clusters compared by the view.

The other mentioned paper [28] is based on organizing the parameters in a hierarchical structure according to what they measure. Then, it works in a bottom-up fashion, each time executing the DM algorithm on a single view (i.e., a node in the hierarchy). The densities around the points in the embedded spaces of the children of a certain node are used as an input for the DM algorithm applied to that node.

The paper has the following structure. Section 14.2 describes the problem setup. Section 14.3 defines the distance metrics, describes the geometry of the resulting embedded space and explains the clustering and classification method. Section 14.4 demonstrates two applications to real-life data sets of the classification method.

14.2 Problem Setup

Assume that the data set X contains m observations where each observation details the values of l nominal parameters. Thus, X can be seen as a $m \times l$ matrix that contains nominal (i.e., categorical) values. The observations in the data set are not necessarily unrelated and several observations may refer to a single studied item or subject in the analysis. One example for such data sets is exception (i.e., software errors) analysis where several exceptions may relate to a single malfunction, which can be identified by the machine and time of these exceptions.

We begin to examine the data set X by defining the subjects of the analysis and relating each observation to the subject to which it refers. This can be done either by external labeling or by grouping the observation according to the values of (some of) their parameters. We denote the set of all subjects by S and its size by $n = |S|$. For each subject $s \in S$, the set of observations in X that refer to it is denoted by X_s.

We assume there is some relation between the parameter sets. Viewing them as a whole might be biased by the number of parameters relating to each perspective. For example, if there are five parameters describing the software components (e.g., process, class, thread) and two describing the thrown exception (e.g., error type), an analysis based on all these parameters would be biased towards the software perspective. To cope with these situations, we use a multi-view approach to analyze the behavior of the subjects. We divide the parameters of the data set to several perspectives, or views, and analyze each of them separately. This way, we provide a complete, unbiased analysis of the data set from several points of view.

The main goal of the analysis in this paper is to find structures and patterns in the data set. We do this by clustering the subjects to a set of classes in each perspective. An examination of the common categories (i.e., nominal values of the parameters) in each class, from each perspective, provides an understanding of the structure of the data set and the types of subjects in it. Also, one can deduce the relations between the subjects based on these understandings.

14.3 Classification Method

In this section, we present the classification method that is applied to each view (i.e., perspective) separately. We start by constructing the view, as a new data set, according to the subjects in S and the parameters selected for the view. Then, we describe the application of DM using a new distance metric. Finally, the structure of the embedded space and the clustering method applied to it are described.

14.3.1 Construction of a View

The analysis begins with selecting the parameters, from the original data set, to be used in the current view. Each observation in X combines nominal values of these parameters. Each subject $s \in S$ is related to some observations in X and so it is described by several combinations of values of the selected parameters. We will denote the set of these combinations for a subject $s \in S$ by V_s. We denote the set of all such combinations in the data set by $V = \cup_{s \in S} V_s$ and their number by $d = |V|$. From this point on, when we refer to combinations of parameter values, or just combinations, we mean the described combinations in V (or V_s for a subject s), unless specifically stated otherwise.

A single view is described by the subjects in S, the combinations in V, and the relations between them. We suggest two approaches to describe and handle this information: the boolean approach and the counter approach. We will describe them side by side in this paper. The boolean approach describes the relation between a subject $s \in S$ and a combination $v \in V$ by a boolean value stating whether or not v was reported for s in the data set, i.e. $v \in V_s$. The counter approach adds the information of how many times this combination reported for s. This describes the relation by a number that counts the observations related to s that contain the combination v.

Formally, each approach constructs an $n \times d$ matrix that describes the view. Each row in this matrix corresponds to a subject in $s \in S$ and each column corresponds to a combination $v \in V$. The boolean approach constructs the matrix B where each cell is defined as

$$[B]_{sv} = b(s, v) \triangleq \begin{cases} 1, & v \in V_s, \\ 0, & v \notin V_s, \end{cases} \quad s \in S,\ v \in V. \tag{14.1}$$

The counter approach constructs the matrix C where each cell is defined as

$$[C]_{sv} = c(s,v) \triangleq \begin{cases} |\{x \in X_s | x \sim v\}|, & v \in V_s, \\ 0, & v \notin V_s, \end{cases} \quad s \in S,\ v \in V, \tag{14.2}$$

where an observation $x \in X$ is similar to a combination $v \in V$ only if this combination of parameter values appears in x. For a subject $s \in S$, we denote its row in B by $b_s = b(s,\cdot)$ and its row in C by $c_s = c(s,\cdot)$. The constructed matrices provide a suitable presentation of the subjects for the analysis from the desired perspective, and whichever of the two described approaches we choose, we will refer to the constructed matrix for that approach as the current view's data set or simply the current view.

Finally, since the methodology we use for analyzing the current view is based upon the distances between data points, we must define a suitable distance metric between subjects for each approach. For the boolean approach, we define the following distance metric between the rows that represents two subjects $s, t \in S$ in the matrix B:

$$\|b_s - b_t\|_b \triangleq \frac{\sum_{v \in V} [b(s,v) \oplus b(t,v)]}{\sum_{v \in V} [b(s,v) \vee b(t,v)]}, \tag{14.3}$$

where the logical operators treat 1 and 0 as *true* and *false*, respectively, and the summation (and division) treat them as numbers. The counter approach defines the following distance metric between the rows that represent two subjects $s, t \in S$ in the matrix C:

$$\|c_s - c_t\|_c \triangleq \frac{\sum_{v \in V} |c(s,v) - c(t,v)|}{\sum_{v \in V} |c(s,v) + c(t,v)|}. \tag{14.4}$$

Both metrics measure the difference between the combinations related to the subjects s and t relative to the total number of combinations reported for any of them. They are similar to the *Jaccard Similarity Coefficient* [23] and the *Tanimoto distance* [29], which are used to compute the similarity and diversity between two data sets.

The rest of the analysis, which is presented in the next sections, does not depend on which approach we use and so we define general notations that will refer to the selected approach. B and C denote the constructed boolean and binary matrix, respectively, denoted by U. We denote by u_s the row of this matrix that represents the subject $s \in S$ (i.e., u_s is b_s or c_s depending on the selected approach). The distance between the rows of U, which represents two subjects $s, t \in S$, according to the selected approach metric, is denoted by $\|u_s - u_t\|_u$. With these notations and the represented constructed view, we are ready to apply the DM to this view and to analyze the lower dimensional representation provided by it.

14.3.2 Application of DM

The DM method analyzes the view's data set by exploring its geometry [8]. It is based on defining the isotropic kernel

$$k_\varepsilon(s,t) \triangleq e^{-\frac{\|u_s - u_t\|_u}{\varepsilon}}, \tag{14.5}$$

where $s,t \in S$ are two subjects and ε is a meta-parameter of the algorithm. This kernel represents the affinities between the two subjects from the perspective of the current view.

The kernel may be viewed as a construction of a weighted graph over the view. The subjects are used as vertices and the weights of the edges are defined by the kernel k_ε. The degree of each subject (i.e., vertex) $s \in S$ in this graph is

$$q_\varepsilon(s) \triangleq \sum k_\varepsilon(s,t). \tag{14.6}$$

Normalizing the kernel with this degree produces an $n \times n$ row stochastic transition matrix M whose cells are $[P]_{st} = p(s,t) = k_\varepsilon(s,t)/q_\varepsilon(s)$, $s,t \in S$, which defines a Markov process (i.e., a diffusion process) over the subjects.

The dimensionality reduction achieved by this diffusion process is a result of spectral analysis of the diffusion kernel. Thus, it is preferable to work with a symmetric conjugate to P that we denote by A and its cells are

$$[A]_{st} = a(s,t) = \frac{k_\varepsilon(s,t)}{\sqrt{q_\varepsilon(s)}\sqrt{q_\varepsilon(t)}} = \sqrt{q_\varepsilon(s)}\,p(s,t)\frac{1}{\sqrt{q_\varepsilon(t)}}, \quad s,t \in S. \tag{14.7}$$

The eigenvalues $1 = \lambda_0 \geq \lambda_1 \geq \ldots$ of A and their corresponding eigenvectors ϕ_i, $i = 0,1,\ldots$, are used to obtain the desired dimensionality reduction by mapping each subject s onto the point $\Phi(s) = (\lambda_i \phi_i(s))_{i=0}^{\delta}$ for a sufficiently small δ, which depends on the decay of the spectrum of A [8, 26]. This construction is also known as the Laplacian of the graph constructed by the kernel [7]. We denote the resulting low-dimensional vector representing a subject s by $\tilde{u}_s = \Phi(s)$, and the set of all such vectors by $\tilde{U}$. We also use the notations $\tilde{b}_s$ (and $\tilde{B}$) or $\tilde{c}_s$ (and $\tilde{C}$), when referring specifically to the boolean approach or the counter approach, respectively.

14.3.3 Construction of the Classes

In practice, for most data sets of the form dealt in this paper (see Sect. 14.2), the vectors in $\tilde{U}$ will have a unique geometry. They form rays emanating from a common center near the origin. This property is due to the discrete nature of the distance metric we used (14.3) or (14.4) and, in turn, the Gaussian kernel (14.5) and the diffusion kernel (14.7) constructed by it. Indeed, the inner product of two vectors $\tilde{u}_s, \tilde{u}_t \in \tilde{U}$

in the embedded space is

$$\langle \tilde{u}_s, \tilde{u}_t \rangle = \left\langle \Phi(s), \Phi(t) \right\rangle = \sum_{i=0}^{\delta} \lambda_i^2 \phi_i(s) \phi_i(t). \tag{14.8}$$

We recall that the eigenvalues of A^2 are $\lambda_0^2, \lambda_1^2, \lambda_2^2, \ldots$ [8, 26]. Thus, according to the spectral theorem and the fast decay of the eigenvalues of A, we get

$$\langle \tilde{u}_s, \tilde{u}_t \rangle = \sum_{i=0}^{\delta} \lambda_i^2 \phi_i(s) \phi_i(t) \approx a^2(s,t) = \left[A^2\right]_{st}. \tag{14.9}$$

Therefore, a small discrete set of values taken by the diffusion kernel leads to a small discrete set of inner products, which determines the angles between the vectors in the embedded space. Since there is a small variety of angles in the embedded space, similar vectors have approximately the same directions (from the origin) and unrelated ones have relatively wide angles between them. This approach is related to cosine similarity, which uses the cosine of the angle between two vectors to define the similarity between them, in the embedded space. The cosine similarity is used to compare documents in text mining [30] and to measure the similarity between clusters.

An examination of the used distance metrics presents a possible explanation for the described structure of the kernel. In both approaches (14.3) and (14.4), totally unrelated subjects have a distance of 1 between them while completely correlated ones have a 0 distance. The range of possible values between 0 and 1 (for two compared subjects) depends on the number of combinations reported for the compared subjects. As more combinations are reported for them, it leads to more possible values. In many cases, however, the maximal number of combinations reported for a single subject is no more than a few dozen combinations while the common number of them for a single subject is less than a dozen. Therefore, the range of possible values for the distance between two subjects is, in practice, fairly limited.

The geometry of the embedded space suggests a new clustering method to be applied to it. Instead of measuring density in Cartesian coordinates, we measure it in polar coordinates. Specifically, the vectors are clustered in this space according to their angle coordinates. First, we find the dominant directions of the rays where large concentrations of vectors lie. Then, we associate each vector with the closest ray. This method yields a set of classes, each of which contains vectors representing similar (i.e. correlated) subjects in the original data set of the currently analyzed view.

One issue that should be pointed out is the concentration of some points near the origin. The embedding process preserves only the principal components of the data. There are many cases in which some of the subjects are completely unrelated to any other subject in the view. Such data points should have a negligible affinity to every other data point; therefore they have an inner product of approximately 0 with all other vectors in $\tilde{U}$. If the dimensionality of the embedded space was large

enough we would see such vectors as almost orthogonal to all the other vectors, but since we deliberately use a low-dimensional embedded space, only their projection on this space is seen. The projections of such data points are thus seen as very close to the origin as they are almost orthogonal to the observed space. Therefore, before applying our analysis we clear a dense area around the origin, which contains all the unrelated vectors to the observed space.

The vectors in the dense central area can be further explored in the same way as the original view. A second iteration might reveal some correlations between the subject corresponding to the vectors in the central area, which were masked by the rest of the vectors in the first one. This would specially be the case if the dimension of the embedded space in the first iteration was too low to encompass the nature of the examined view. If, on the other hand, it was sufficient to represent the view, the next iteration would show a clutter of uncorrelated vectors with no apparent relations between them.

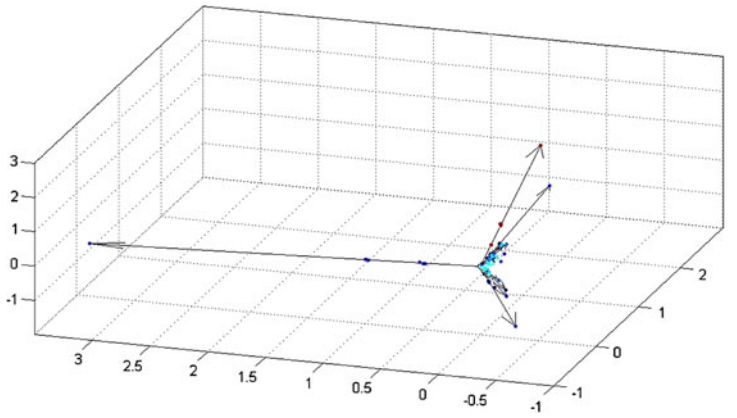
14.4 Empirical Results

In this section, we present two applications of the polar classification method for analyzing real-life data sets. The first example demonstrates the usage of this method to classify malfunctions from an error monitoring log. The second example shows tools for supporting management decisions during the testing phases (i.e., QA cycles) of a software development process that is based on the polar classification method.

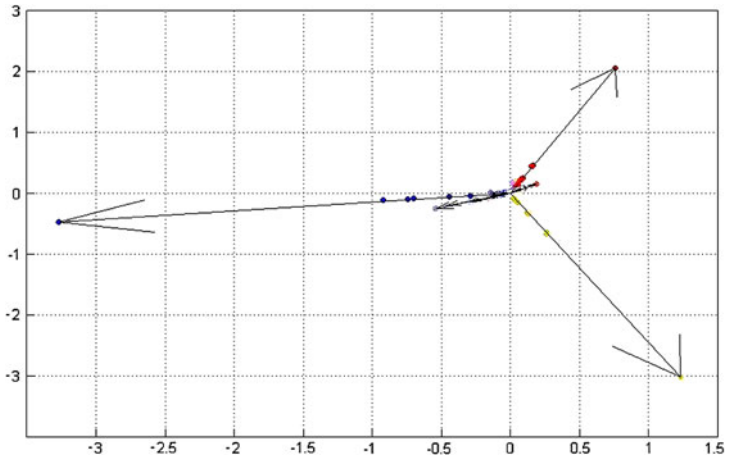
14.4.1 Error Monitoring

We applied the polar classification method to a data set that contains a log of errors that were recorded by a wide-scale distributed system. Each entry in the log records information about the malfunctioning server, the time of the error and the details of the error. We used the polar clustering and classification method to classify distinct events, which are identified by a server name and the time of the event according to the components that reported the malfunctions.

In order to achieve the desired clustering and classification we used the boolean approach to construct a 4018×719 flag matrix, which indicates the components that were malfunctioning in each event (i.e., specific server and time). Each row in this matrix corresponds to a single event and each column corresponds to a single component. Thus, there were 4018 distinct events and 719 distinct components in the analyzed log. Next, we constructed the boolean distance metric (14.3) between rows of this flag matrix and applied the DM method according to the calculated distances. We used the first 20 eigenvectors of the diffusion kernel (defined by (14.7)). Thus, our embedded space was 20-dimensional. This space is illustrated in Fig. 14.1.



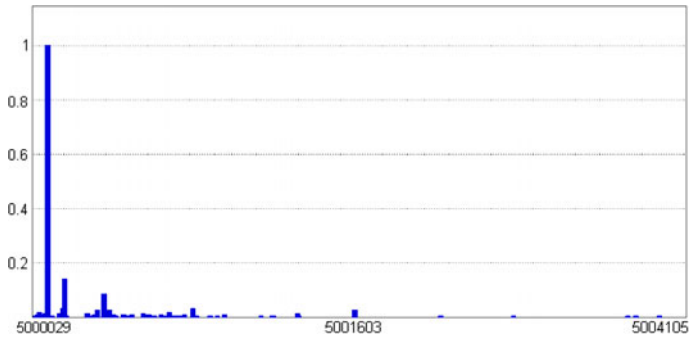
(a) First three axes of the embedded space



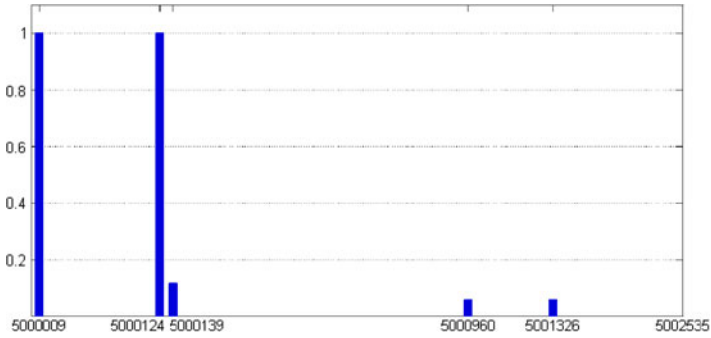
(b) Axes 7 and 8 of the embedded space

Fig. 14.1 An illustration of the 20-dimensional embedded space. The points correspond to the examined events, and they are colored according to the detected clusters (i.e., rays). The vectors display the detected rays on which the points are concentrated

The events in the embedded space form distinct rays emanating from a mutual central point. These rays were detected and the events were classified according to the ray on which they lie. Next, we examined the events in each of the resulting classes. Every class had a few components that were reported in almost all of the events in the class. We refer to such components as the dominant components of the examined class. The bar plot in Fig. 14.2(a) demonstrates a class with a single dominant component and the one in Fig. 14.2(b) shows an example of a class with two dominant components. Each bar in these plots represents a single component. The height of the bar indicates how many of the events, in the examined class, reported it.



(a) A class with one dominant component (DB adapter in this case)

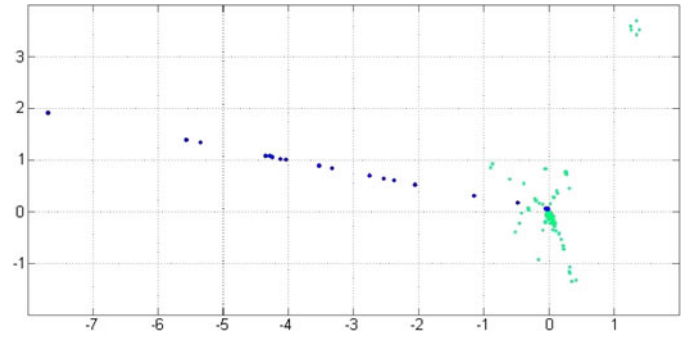


(b) A class with two dominant components (input fetcher and data parser in this case)

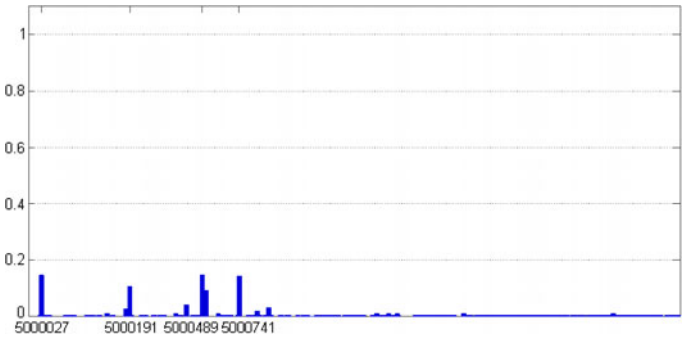
Fig. 14.2 Examples of classes with one and two dominant components. The bar plot shows for each component how many of the events in this class reported it (e.g., 0 means none and 1 means all)

The embedded space in this case also had a dense central area, which contains 1539 events (out of the original 4018) that were unrelated to the detected classes. We examined this central area by applying the same analysis to the events in it. The classes that resulted from this analysis showed more subtle patterns than these from the first iteration. An example of such a pattern is shown in Figs. 14.3, 14.4, and 14.5, which presents three sections of a single class (i.e., ray).[1] When all the events in the class are considered (Fig. 14.3(a)), the dominant components of the class are not apparent (Fig. 14.3(b)). If, on the other hand, we only consider events, which are very far away from the central area (Fig. 14.4(a)), then only five components, which are reported for all of these events, are left as dominant (Fig. 14.4(b)). Finally, by filtering out only the events that are very close to the central area and

[1]The dominant components are clear when points that are too close to the central area are not considered. The dominant components in this case have various interrelated functions specific to the analyzed system.



(a) Eigenvectors 5 and 10 of the embedded space of the second iteration



(b) Dominant components

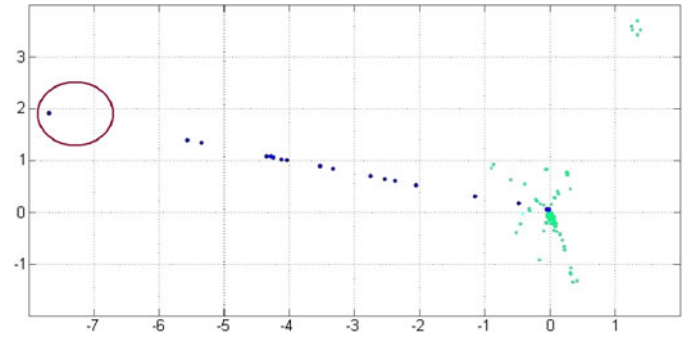
Fig. 14.3 A class with a few dominant components in the second iteration: An entire class

considering the remaining events (Fig. 14.5(a)), we get the bar plot in Fig. 14.5(b), which still indicates about three dominant components of this class and two less dominant ones.

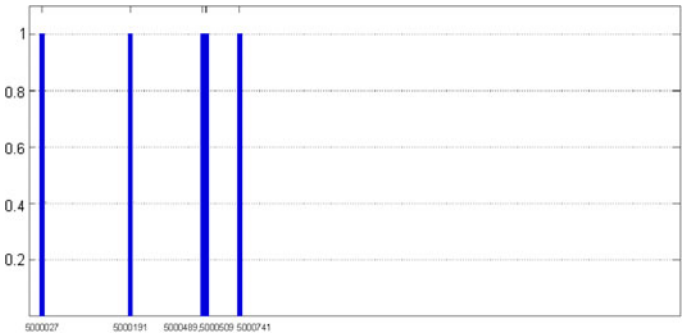
We used 25 dominant classes from each of the conducted iterations. For each class we identified its dominant components. Therefore, the original 4018 events were clustered into 50 different classes that covered 3292 events. The remaining 726 events lied in the central area of the embedded space of the second iteration and did not show any special correlations. These results provide vital information about the common combinations of malfunctioning components, which cover over 75 % of the events in the log. This information can be used both for root-cause analysis in order to find the programmatic defects that cause these problems and as a guidance tool for the development of future versions of the system.

14.4.2 Quality Assurance

The term quality assurance (QA) in software development refers to a phase in the development life cycle, in which the developed product is tested for inherent oper-



(a) Eigenvectors 5 and 10 of the embedded space of the second iteration

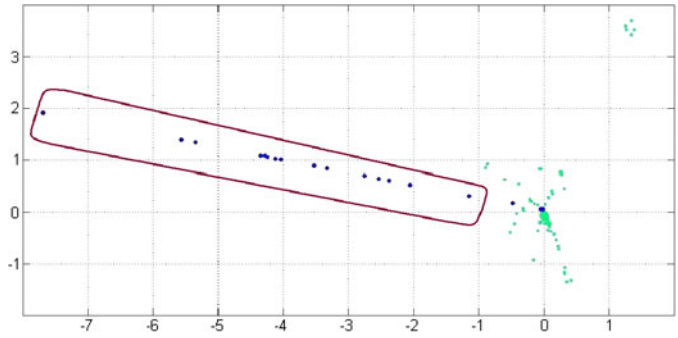


(b) Dominant components

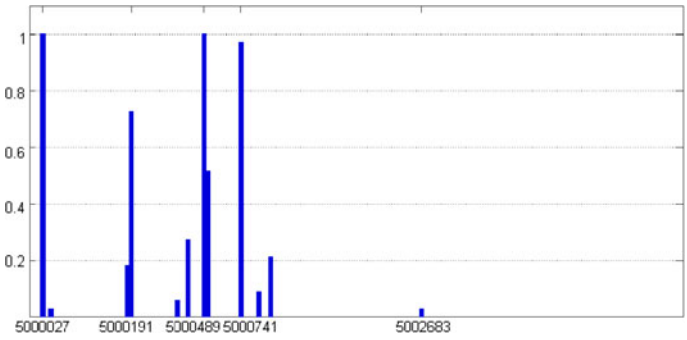
Fig. 14.4 A class with a few dominant components in the second iteration: The farthest points from the central area

ational errors. Usually, several testing cycles are applied and the defects found in each cycle are fixed before the next cycle begins. In most development teams, a defect management tool is used to log all the detected bugs and prioritize them based on their impact and severity. When developing and maintaining multiple configurations and several different versions of the software, it is not trivial to determine the priority of a given bug. In order to do so, potential benefits across all configurations and versions have to be considered. For example, a small set of defects can, in fact, be the root cause of a possible instability in several configurations.

We applied the polar classification method to the analyzing software defects tasks, then classifying the configurations and the versions of a project based on the detected defects. We used a data set that contains information about defects that were detected in several configurations and in several versions. This information was recorded during several testing cycles of a software project. The detail that were recorded for each defect are the defected features, the defect type, its detection, the configuration on which it occurred and the software version in which it occurred.



(a) Eigenvectors 5 and 10 of the embedded space of the second iteration

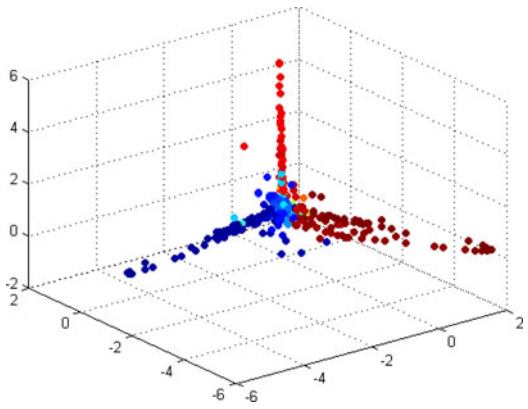


(b) Dominant components

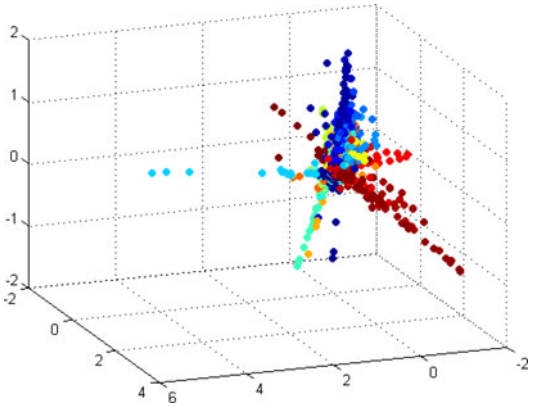
Fig. 14.5 A class with a few dominant components in the second iteration: Points not too close to the central area

We defined the subjects of the analysis, in this case, as pairs of the version (i.e., cycle) and configuration of the software. They are clustered by the defected features detected in each of these pairs. We used the counter approach to construct a counter matrix, where each row corresponds to a unique pair of version and configuration and each column corresponds to a single software feature. The data set contained 1426 distinct version-configuration pairs and 2275 different defected features (i.e., the counter matrix was a 1426×2275 matrix). Then, we used the counter distance metric (14.4) to compute the distances between rows of the counter matrix where the DM method is applied based on the computed distances. The resulting 20-dimensional embedded space is illustrated in Fig. 14.6.

The embedded space in this example is similar in shape to the one in the previous example. Again, the embedded data points, which correspond to version-configuration pairs, are organized on rays emanating from a mutual center. The version-configuration pairs are clustered according to the rays on which they lie in the embedded space. We performed a second iteration of the analysis, similar to the one explained in the previous example, on those that lie in the central area (i.e., the points that are unrelated to any cluster). In the first iteration, 981 of the 1426



(a) First three eigenvectors of the embedded space

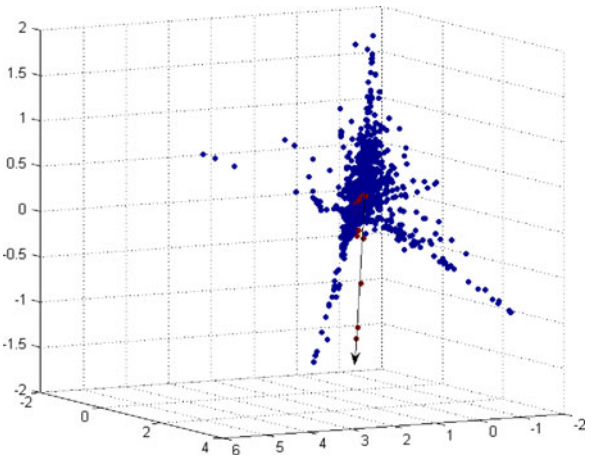


(b) Eigenvectors 4, 5, and 6 of the embedded space

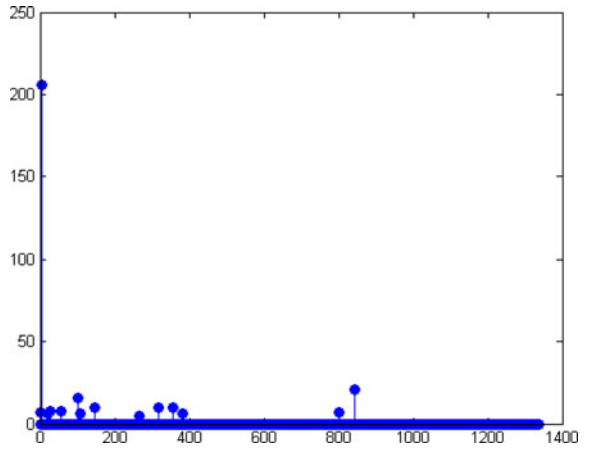
Fig. 14.6 An illustration of the 20-dimensional embedded space. The points correspond to the software versions. They are colored according to the detected clusters (i.e., rays)

data points were clustered into 86 different clusters. The second iteration, which was performed using the remaining 535 data points, detected 30 additional clusters that encompassed 219 data points (out of the 535 analyzed). Overall, 1207 version-configuration pairs were clustered into 116 classes, leaving 219 unclassified pairs, which were not correlated with the rest of the pairs.

The detected classes show a situation similar to the one in the previous example; i.e., each class has a small set of dominant defected features that occur in almost every version-configuration pair in the class. Figure 14.7 demonstrates this result by showing a single class with one dominant defected feature reported by most of the data points in it. The achieved versions and configuration clustering by using the polar classification method, can be used to find similar behaviors between different setting in the system. Defects (specifically defected features) can be prioritized according to the classes in which they are detected. There are classes that contain many configuration and many versions should indicate wide and long standing problems.



(a) A class with one dominant component



(b) A dominant component of the separated class

Fig. 14.7 An example of a class with one dominant component. The bar plot shows the sum of reports made by all software versions in this class for each component

14.5 Conclusions

We presented a distance metric that utilizes the DM methodology for analyzing nominal data sets. We used a multi-view approach to analyze a data set from several perspectives instead of examining it as a whole. From each perspective, a diffusion kernel was constructed and the analyzed items of the perspective were mapped to Euclidean space using spectral analysis of this kernel. The embedded items formed rays in the embedded space that were emanating from a common central area, which is the new origin. These rays indicate a dominant pattern in the data set, and can be used to cluster and classify the analyzed items. The results of this clustering can also be used as a basis for further analysis of the data, e.g., by further analyzing the similarities between the clusters and rating each of them according to its impact on the other clusters.

References

1. Agrawal R, Gehrke J, Gunopulos D, Raghavan P (1998) Automatic subspace clustering of high dimensional data for data mining applications. In: SIGMOD '98: proceedings of the 1998 ACM SIGMOD international conference on management of data. ACM, New York, pp 94–105
2. Ankerst M, Breunig MM, Kriegel HP, Sander J (1999) OPTICS: ordering points to identify the clustering structure. In: SIGMOD '99: proceedings of the 1999 ACM SIGMOD international conference on management of data. ACM, New York, pp 49–60
3. Babuška R (1998) Fuzzy modeling for control. Kluwer, Norwell
4. Berkhin P (2006) A survey of clustering data mining techniques. Grouping Multidimensional Data Cl(c):25–71
5. Bickel S, Scheffer T (2004) Multi-view clustering. In: ICDM '04: proceedings of the fourth IEEE international conference on data mining. IEEE, Washington, pp 19–26
6. Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on computational learning theory, Madison, WI, 1998. ACM, New York, pp 92–100
7. Chung F (1997) Spectral graph theory. CBMS regional conference series in mathematics, vol 92. AMS, Providence
8. Coifman RR, Lafon S (2006) Diffusion maps. Appl Comput Harmon Anal 21(1):5–30
9. Dasgupta S, Littman ML, McAllester D (2001) PAC generalization bounds for co-training. Technical report, AT&T Labs-Research
10. David G (2009) Anomaly detection and classification via diffusion processes in hyper-networks. PhD thesis, School of Computer Science, Tel Aviv University
11. David G, Averbuch A (2012) Hierarchical data organization, clustering and denoising via localized diffusion folders. Appl Comput Harmon Anal 33(1):1–23
12. David G, Averbuch A (2011) Localized diffusion. Part II: Coarse-grained process (submitted)
13. David G, Averbuch A (2012) SpectralCAT: categorical spectral clustering of numerical and nominal data. Pattern Recognit 45(1):416–433
14. de Diego IM, Munoz A, Moguerza J (2010) Methods for the combination of kernel matrices within a support vector framework. Mach Learn 78:137–174
15. de Sa VR, Gallagher PW, Lewis JM, Malave VL (2010) Multi-view kernel construction. Mach Learn 79(1):47–71
16. Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD '96: proceedings of the 2nd international conference on knowledge discovery and data mining. AAAI, New York, pp 226–231
17. Everitt B, Landau S, Leese M (2001) Cluster analysis, 4th edn. Arnold, London
18. Guha S, Rastogi R, Shim K (1998) CURE: an efficient clustering algorithm for large databases. In: SIGMOD '98: proceedings of the 1998 ACM SIGMOD international conference on management of data. ACM, New York, pp 73–84
19. Guha S, Rastogi R, Shim K (2000) ROCK: a robust clustering algorithm for categorical attributes. Inf Syst (Oxf) 25(5):345–366
20. Hinneburg A, Keim DA (1998) An efficient approach to clustering in large multimedia databases with noise. In: KDD '98: proceedings of the 4th international conference on knowledge discovery and data mining, pp 58–65
21. Huang Z (1997) A fast clustering algorithm to cluster very large categorical data sets in data mining. In: SIGMOD-DMKD '97: workshop on research issues on data mining and knowledge discovery
22. Huang Z (1998) Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min Knowl Discov 2(3):283–304
23. Jaccard P (1901) Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull Soc Vaud Sci Nat 37:547–579
24. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323

25. Karypis G, Han EH, Kumar V (1999) Chameleon: hierarchical clustering using dynamic modeling. Computer 32(8):68–75
26. Lafon S (2004) Diffusion maps and geometric harmonics. PhD thesis, Yale University
27. MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley symposium on mathematical statistics and probability. Statistics, vol I. Univ California Press, Berkeley, pp 281–297
28. Rabin N (2010) Data mining dynamically evolving systems via diffusion methodologies. PhD thesis, School of Computer Science, Tel Aviv University
29. Rogers DJ, Tanimoto TT (1960) A computer program for classifying plants. Science 132(3434):1115–1118
30. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Process Manag 24(5):513–523
31. Sebban M, Nock R (2002) A hybrid filter/wrapper approach of feature selection using information theory. Pattern Recognit 35(4):835–846
32. Sheikholeslami G, Chatterjee S, Zhang A (2000) WaveCluster: A wavelet-based clustering approach for spatial data in very large databases. VLDB J 8(3–4):289–304
33. Stanfill C, Waltz D (1986) Toward memory-based reasoning. Commun ACM 29(12):1213–1228
34. Strehl A, Ghosh J (2000) A scalable approach to balanced, high-dimensional clustering of market-baskets. In: HiPC '00: proceedings of the 7th international conference on high performance computing. Springer, London, pp 525–536
35. Wang K, Xu C, Liu B (1999) Clustering transactions using large items. In: CIKM '99: proceedings of the 8th international conference on information and knowledge management. ACM, New York, pp 483–490
36. Wang P (2008) Clustering and classification techniques for nominal data application. PhD thesis, City University of Hong Kong
37. Wang W, Yang J, Muntz R (1997) STING: a statistical information grid approach to spatial data mining. In: VLDB '97: proceedings of the 23rd international conference on very large data bases. Morgan Kaufmann, San Francisco, pp 186–195
38. Wang W, Yang J, Muntz R (1999) STING+: an approach to active spatial data mining. In: ICDE '99: proceedings of the 15th international conference on data engineering. IEEE, Los Alamitos, pp 116–125
39. Yang Y, Guan X, You J (2002) CLOPE: a fast and effective clustering algorithm for transactional data. In: KDD '02: proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 682–687
40. Yarowsky D (1995) Unsupervised word sense disambiguation rivaling supervised methods. In: ACL '95: proceedings of the 33rd annual meeting on association for computational linguistics. Association for Computational Linguistics, Stroudsburg, pp 189–196
41. Yun CH, Chuang KT, Chen MS (2001) An efficient clustering algorithm for market basket data based on small large ratios. In: COMPSAC '01: proceedings of the 25th international computer software and applications conference on invigorating software development. IEEE, Washington, pp 505–510
42. Zhang T, Ramakrishnan R, Livny M (1996) BIRCH: an efficient data clustering method for very large databases. In: SIGMOD '96: proceedings of the 1996 ACM SIGMOD international conference on management of data. ACM, New York, pp 103–114
43. Zhao Y, Song J (2001) GDILC: a grid-based density-isoline clustering algorithm. In: ICII '01: proceedings of the international conferences on info-tech and info-net, vol 3. IEEE, New York, pp 140–145

Part IV
Optimization Methods

Chapter 15
Subgradient and Bundle Methods for Nonsmooth Optimization

Marko M. Mäkelä, Napsu Karmitsa, and Adil Bagirov

Abstract The nonsmooth optimization methods can mainly be divided into two groups: subgradient and bundle methods. Usually, when developing new algorithms and testing them, the comparison is made between similar kinds of methods. The goal of this work is to test and compare different bundle and subgradient methods as well as some hybrids of these two and/or some others. The test set included a large amount of different unconstrained nonsmooth minimization problems, e.g., convex and nonconvex problems, piecewise linear and quadratic problems, and problems with different sizes. Rather than foreground some method over the others, our aim is to get some insight on which method is suitable for certain types of problems.

15.1 Introduction

We consider unconstrained nonsmooth optimization (NSO) problems of the form

$$\min_{x \in \mathbb{R}^n} f(x), \tag{15.1}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is supposed to be locally Lipschitz continuous. Note that no differentiability or convexity assumptions are made.

NSO problems of type (15.1) arise in many application areas: in economics [38], mechanics [37], engineering [36], control theory [11], optimal shape design [17], data mining [1, 7] and in particular cluster analysis [12], and machine learning [20].

M.M. Mäkelä (✉) · N. Karmitsa
Department of Mathematics, University of Turku, 20014 Turku, Finland
e-mail: makela@utu.fi

N. Karmitsa
e-mail: napsu@karmitsa.fi

A. Bagirov
Centre for Informatics and Applied Optimization, School of Science, Information Technology and Engineering, University of Ballarat, University Drive, Mount Helen, PO Box 663, Ballarat, VIC 3353, Australia
e-mail: a.bagirov@ballarat.edu.au

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_15,

Most of the methods for solving problems of type (15.1) can be divided into two main groups: subgradient (see, e.g., [4, 5, 42, 43]) and bundle methods (see, e.g., [14, 18, 23, 32, 35, 40, 41]). Both of these method groups have their own supporters. Usually, when developing new methods, researchers compare them with similar methods. Moreover, it is quite common that the test set used is rather concise.

In this work, we compare different subgradient and bundle methods, as well as some of the methods that lie between these two. The main criteria in numerical comparison are the efficiency and the reliability of the methods. Moreover, we use a broad test setting including different classes of nonsmooth problems. All the solvers tested are so-called general black box methods and, naturally, cannot beat the codes designed specially for a particular class of problems (say, e.g., for piecewise linear, min-max, or partially separable problems). However, rather than seeing this generality as a weakness, it should be seen as a strength due to the minimal information of the objective function required for the calculations. Namely, the value of the objective function and, possibly, one arbitrary subgradient (the generalized gradient [10]) at each point.

The aim of our research is not to foreground some method over the others—it is a well-known fact that different methods work well for different types of problems and none of them is good for all types of problems—but to get some kind of insight on which kind of method to select for certain types of problems.

This work is organized as follows. Section 15.2 introduces the NSO methods tested and compared. The results of the numerical experiments are presented and discussed in Sects. 15.3 and 15.4 concludes the work and gives our credentials for well-performing algorithms for different problem classes.

In what follows, we denote by $\|\cdot\|$ the Euclidean norm in $\mathbb{R}^n$ and by $\mathbf{a}^T\mathbf{b}$ the inner product of the vectors $\mathbf{a}$ and $\mathbf{b}$. The *subdifferential* $\partial f(\mathbf{x})$ [10] of a locally Lipschitz continuous function $f : \mathbb{R}^n \to \mathbb{R}$ at any point $\mathbf{x} \in \mathbb{R}^n$ is given by

$$\partial f(\mathbf{x}) = \operatorname{conv}\left\{\lim_{i\to\infty} \nabla f(\mathbf{x}_i) \mid \mathbf{x}_i \to \mathbf{x} \text{ and } \nabla f(\mathbf{x}_i) \text{ exists}\right\},$$

where "conv" denotes the convex hull of a set. Each vector $\boldsymbol{\xi} \in \partial f(\mathbf{x})$ is called a *subgradient*.

15.2 Methods

In this section, we give short descriptions of the methods to be compared. For more details we refer to [22] and to the original references. In what follows (if not stated otherwise), we assume that at every point $\mathbf{x}$ we can evaluate the value of the objective function $f(\mathbf{x})$ and an arbitrary subgradient $\boldsymbol{\xi}$ from the subdifferential $\partial f(\mathbf{x})$.

15.2.1 Standard Subgradient Method

The first method to be considered here is the cornerstone of NSO: the standard subgradient method [42]. The idea behind subgradient methods (Kiev methods) is to generalize smooth methods (e.g., the steepest descent method) by replacing the gradient with an arbitrary subgradient. Therefore, the iteration formula for these methods is

$$\mathbf{x}_{k+1} = \mathbf{x}_k - t_k \frac{\boldsymbol{\xi}_k}{\|\boldsymbol{\xi}_k\|},$$

where $\boldsymbol{\xi}_k \in \partial f(\mathbf{x}_k)$ is any subgradient and $t_k > 0$ is a predetermined step size.

Due to this simple structure and low storage requirements, subgradient methods are widely used methods in NSO. However, basic subgradient methods suffer from some serious disadvantages: a nondescent search direction may occur and thus, the selection of step size is difficult; there exists no implementable subgradient-based stopping criterion; and the convergence speed is poor (not even linear) (see, e.g., [26]).

The standard subgradient method is proved to be globally convergent if the objective function is convex and the step sizes satisfy

$$\lim_{k\to\infty} t_k = 0 \quad \text{and} \quad \sum_{j=1}^{\infty} t_j = \infty.$$

15.2.2 Shor's r-Algorithm (Space Dilation Method)

Next we shortly describe the ideas of a more sophisticated subgradient method, the well-known Shor's r-algorithm with space dilations along the difference of two successive subgradients. The basic idea of Shor's r-algorithm is to interpolate between the steepest descent and conjugate gradient method.

The iteration formula for Shor's r-algorithm is

$$\mathbf{x}_{k+1} = \mathbf{x}_k - t_k B_{k+1} B_{k+1}^T \boldsymbol{\xi}_k,$$

where $\boldsymbol{\xi}_k \in \partial f(\mathbf{x}_k)$ and $t_k > 0$. The space dilation matrix B_{k+1} is initialized with $B_1 = I$ and it is updated by

$$B_{k+1} = B_k\big(I + (\beta - 1)\mathbf{s}_k \mathbf{s}_k^T\big),$$

where $\beta \in (0, 1)$, $\mathbf{s}_k = \mathbf{r}_k / \|\mathbf{r}_k\|$ and $\mathbf{r}_k = B_k^T(\boldsymbol{\xi}_k - \boldsymbol{\xi}_{k-1})$.

In order to turn the above r-algorithm into an efficient optimization routine, one has to find a solution to the following problems: how to choose the step sizes t_k (including the initial step size t_1) and how to design a stopping criterion which does not need information on subgradients.

If the objective function is convex and twice continuously differentiable, its Hessian is Lipschitz, and the starting point is chosen from some neighborhood of the optimal solution, then the n-step quadratic rate convergence can be proved for the r-algorithm. If the objective function is nonconvex and coercive under some additional assumptions, then the r-algorithm is convergent to isolated local minimizers [42].

15.2.3 Proximal Bundle Method (PBM)

In this subsection, we describe the ideas of the proximal bundle method (PBM) for nonsmooth and nonconvex minimization (see, e.g., [24, 35, 41]).

The basic idea of bundle methods is to approximate the whole subdifferential of the objective function instead of using only one arbitrary subgradient at each point. In practice, this is done by gathering subgradients from the previous iterations into a bundle. Suppose that at the k-th iteration of the algorithm we have the current iteration point $\mathbf{x}_k$ and some trial points $\mathbf{y}_j \in \mathbb{R}^n$ (from past iterations) and subgradients $\boldsymbol{\xi}_j \in \partial f(\mathbf{y}_j)$ for $j \in J_k$, where the index set $J_k \neq \emptyset$ and $J_k \subset \{1, \ldots, k\}$.

The idea behind the PBM is to approximate the objective function f below by a piecewise linear function, that is, f is replaced by the so-called *cutting-plane model*

$$\hat{f}_k(\mathbf{x}) = \max_{j \in J_k} \left\{ f(\mathbf{y}_j) + \boldsymbol{\xi}_j^T (\mathbf{x} - \mathbf{y}_j) \right\}. \tag{15.2}$$

This model can be written in an equivalent form

$$\hat{f}_k(\mathbf{x}) = \max_{j \in J_k} \left\{ f(\mathbf{x}_k) + \boldsymbol{\xi}_j^T (\mathbf{x} - \mathbf{x}_k) - \alpha_j^k \right\},$$

where

$$\alpha_j^k = f(\mathbf{x}_k) - f(\mathbf{y}_j) - \boldsymbol{\xi}_j^T (\mathbf{x}_k - \mathbf{y}_j) \quad \text{for all } j \in J_k$$

is a so-called *linearization error*. If f is convex, then $\hat{f}_k$ is an underestimate for f and $\alpha_j^k \geq 0$ for all $j \in J_k$. In the nonconvex case, these facts are not valid anymore and thus the linearization error α_j^k can be replaced by the so-called *subgradient locality measure* (cf. [23])

$$\beta_j^k = \max \left\{ \left| \alpha_j^k \right|, \gamma \| \mathbf{x}_k - \mathbf{y}_j \|^2 \right\}, \tag{15.3}$$

where $\gamma \geq 0$ is the *distance measure parameter* ($\gamma = 0$ if f is convex). Then obviously $\beta_j^k \geq 0$ for all $j \in J_k$ and $\min_{\mathbf{x} \in K} \hat{f}_k(\mathbf{x}) \leq f(\mathbf{x}_k)$.

The descent direction is calculated by

$$\mathbf{d}_k = \arg\min_{\mathbf{d} \in \mathbb{R}^n} \left\{ \hat{f}_k(\mathbf{x}_k + \mathbf{d}) + \frac{1}{2} u_k \mathbf{d}^T \mathbf{d} \right\}, \tag{15.4}$$

where the stabilizing term $\frac{1}{2}u_k\mathbf{d}^T\mathbf{d}$ guarantees the existence of the solution $\mathbf{d}_k$ and keeps the approximation local enough. The weighting parameter $u_k > 0$ improves the convergence rate and accumulates some second order information about the curvature of f around $\mathbf{x}_k$ (see, e.g., [24, 35, 41]).

In order to determine the step size into the search direction $\mathbf{d}_k$, the PBM uses the following *line search procedure*: Assume that $m_L \in (0, \frac{1}{2})$, $m_R \in (m_L, 1)$ and $\bar{t} \in (0, 1]$ are some fixed line search parameters. We first search for the largest number $t_L^k \in [0, 1]$ such that $t_L^k \geq \bar{t}$ and

$$f\left(\mathbf{x}_k + t_L^k\mathbf{d}_k\right) \leq f(\mathbf{x}_k) + m_L t_L^k v_k, \tag{15.5}$$

where v_k is the predicted amount of descent

$$v_k = \hat{f}_k(\mathbf{x}_k + \mathbf{d}_k) - f(\mathbf{x}_k) < 0.$$

If such a parameter exists, we take a *long serious step*

$$\mathbf{x}_{k+1} = \mathbf{x}_k + t_L^k\mathbf{d}_k \quad \text{and} \quad \mathbf{y}_{k+1} = \mathbf{x}_{k+1}. \tag{15.6}$$

Otherwise, if (15.5) holds but $0 < t_L^k < \bar{t}$, we take a *short serious step*

$$\mathbf{x}_{k+1} = \mathbf{x}_k + t_L^k\mathbf{d}_k \quad \text{and} \quad \mathbf{y}_{k+1} = \mathbf{x}_k + t_R^k\mathbf{d}_k$$

and, if $t_L^k = 0$, we take a *null step*

$$\mathbf{x}_{k+1} = \mathbf{x}_k \quad \text{and} \quad \mathbf{y}_{k+1} = \mathbf{x}_k + t_R^k\mathbf{d}_k, \tag{15.7}$$

where $t_R^k > t_L^k$ is such that

$$-\beta_{k+1}^{k+1} + \boldsymbol{\xi}_{k+1}^T\mathbf{d}_k \geq m_R v_k. \tag{15.8}$$

In short serious steps and null steps there exists discontinuity in the gradient of f. Then the requirement (15.8) ensures that $\mathbf{x}_k$ and $\mathbf{y}_{k+1}$ lie on the opposite sides of this discontinuity and the new subgradient $\boldsymbol{\xi}_{k+1} \in \partial f(\mathbf{y}_{k+1})$ will force a remarkable modification of the next search direction finding problem. The iteration is terminated if $v_k \geq -\varepsilon_s$, where $\varepsilon_s > 0$ is a final accuracy tolerance supplied by the user.

Under the upper semi-smoothness assumption [6] the PBM can be proved to be globally convergent for locally Lipschitz continuous functions (see, e.g., [24, 35]). In addition, in order to implement the above algorithm one has to bound somehow the number of stored subgradient and trial points, that is, the cardinality of the index set J_k. The global convergence of bundle methods with a limited number of stored subgradients can be guaranteed by using a subgradient aggregation strategy [23], which accumulates information from the previous iterations. The convergence rate of the PBM is linear for convex functions [39] and for piecewise linear problems the PBM achieves a finite convergence [41].

15.2.4 Bundle Newton Method (BNEW)

Next we describe the main ideas of the second-order bundle-Newton method (BNEW) [29]. We suppose that at each $\mathbf{x} \in \mathbb{R}^n$ we can evaluate, in addition to the function value and an arbitrary subgradient $\boldsymbol{\xi} \in \partial f(\mathbf{x})$, also an $n \times n$ symmetric matrix $G(\mathbf{x})$ approximating the Hessian matrix $\nabla^2 f(\mathbf{x})$. Now, instead of the piecewise linear cutting-pane model (15.2) we introduce a piecewise quadratic model of the form

$$\tilde{f}_k(\mathbf{x}) = \max_{j \in J_k} \left\{ f(\mathbf{y}_j) + \boldsymbol{\xi}_j^T(\mathbf{x} - \mathbf{y}_j) + \frac{1}{2}\varrho_j(\mathbf{x} - \mathbf{y}_j)^T G_j(\mathbf{x} - \mathbf{y}_j) \right\}, \tag{15.9}$$

where $G_j = G(\mathbf{y}_j)$ and $\varrho_j \in [0, 1]$ is some damping parameter. The model (15.9) can be written equivalently as

$$\tilde{f}_k(\mathbf{x}) = \max_{j \in J_k} \left\{ f(\mathbf{x}_k) + \boldsymbol{\xi}_j^T(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}\varrho_j(\mathbf{x} - \mathbf{x}_k)^T G_j(\mathbf{x} - \mathbf{x}_k) - \alpha_j^k \right\}$$

and for all $j \in J_k$ the linearization error takes the form

$$\alpha_j^k = f(\mathbf{x}_k) - f(\mathbf{y}_j) - \boldsymbol{\xi}_j^T(\mathbf{x}_k - \mathbf{y}_j) - \frac{1}{2}\varrho_j(\mathbf{x}_k - \mathbf{y}_j)^T G_j(\mathbf{x}_k - \mathbf{y}_j). \tag{15.10}$$

Note that now, even in the convex case, α_j^k might be negative. Therefore we replace the linearization error (15.10) by the subgradient locality measure (15.3) and we remain the property $\min_{\mathbf{x} \in \mathbb{R}^n} \tilde{f}_k(\mathbf{x}) \leq f(\mathbf{x}_k)$ (see [29]).

The search direction $d_k \in \mathbb{R}^n$ is now calculated as the solution of

$$\mathbf{d}_k = \arg\min_{\mathbf{d} \in \mathbb{R}^n} \left\{ \tilde{f}_k(\mathbf{x}_k + \mathbf{d}) \right\}. \tag{15.11}$$

The line search procedure of the BNEW follows the same principles than in the PBM (see Sect. 15.2.3). The only remarkable difference occurs in the termination condition for short and null steps. In other words, (15.8) is replaced by two conditions

$$-\beta_{k+1}^{k+1} + \left(\boldsymbol{\xi}_{k+1}^{k+1}\right)^T d_k \geq m_R v_k$$

and

$$\|\mathbf{x}_{k+1} - \mathbf{y}_{k+1}\| \leq C_S,$$

where $C_S > 0$ is a parameter supplied by the user.

Under the upper semi-smoothness assumption [6] the BNEW can be proved to be globally convergent for locally Lipschitz continuous objective functions. For strongly convex functions, the convergence rate of the BNEW is superlinear [29].

15.2.5 Limited Memory Bundle Method (LMBM)

In this subsection, we very shortly describe the limited memory bundle algorithm (LMBM) [15, 16] for solving general, possibly nonconvex, large-scale NSO problems. The method is a hybrid of the variable metric bundle methods [44] and the limited memory variable metric methods (see, e.g., [9]), where the first ones have been developed for small- and medium-scale nonsmooth optimization and the latter ones, on the contrary, for smooth large-scale optimization.

LMBM exploits the ideas of the variable metric bundle methods, namely the utilization of null steps, simple aggregation of subgradients, and the subgradient locality measures, but the search direction $\mathbf{d}_k$ is calculated using a limited memory approach. That is,

$$\mathbf{d}_k = -D_k \tilde{\boldsymbol{\xi}}_k,$$

where $\tilde{\boldsymbol{\xi}}_k$ is an (aggregate) subgradient and D_k is the limited memory variable metric update that, in the smooth case, represents the approximation of the inverse of the Hessian matrix. Note that the matrix D_k is not formed explicitly but the search direction $\mathbf{d}_k$ is calculated using the limited memory approach.

The LMBM uses the original subgradient $\boldsymbol{\xi}_k$ after the serious step (cf. (15.6)) and the aggregate subgradient $\tilde{\boldsymbol{\xi}}_k$ after the null step (cf. (15.7)) for direction finding (i.e. we set $\tilde{\boldsymbol{\xi}}_k = \boldsymbol{\xi}_k$ if the previous step was a serious step). The aggregation procedure is carried out by determining multipliers λ_i^k satisfying $\lambda_i^k \geq 0$ for all $i \in \{1, 2, 3\}$, and $\sum_{i=1}^{3} \lambda_i^k = 1$ that minimize the function

$$\begin{aligned}\varphi(\lambda_1, \lambda_2, \lambda_3) = [\lambda_1 \boldsymbol{\xi}_m + \lambda_2 \boldsymbol{\xi}_{k+1} + \lambda_3 \tilde{\boldsymbol{\xi}}_k]^T D_k [\lambda_1 \boldsymbol{\xi}_m + \lambda_2 \boldsymbol{\xi}_{k+1} + \lambda_3 \tilde{\boldsymbol{\xi}}_k] \\ + 2(\lambda_2 \beta_{k+1} + \lambda_3 \tilde{\beta}_k).\end{aligned}$$

Here $\boldsymbol{\xi}_m \in \partial f(\mathbf{x}_k)$ is the current subgradient (m denotes the index of the iteration after the latest serious step, i.e. $\mathbf{x}_k = \mathbf{x}_m$), $\boldsymbol{\xi}_{k+1} \in \partial f(\mathbf{y}_{k+1})$ is the auxiliary subgradient, and $\tilde{\boldsymbol{\xi}}_k$ is the current aggregate subgradient from the previous iteration ($\tilde{\boldsymbol{\xi}}_1 = \boldsymbol{\xi}_1$). In addition, β_{k+1} is the current subgradient locality measure (cf. (15.3)) and $\tilde{\beta}_k$ is the current aggregate subgradient locality measure ($\tilde{\beta}_1 = 0$). The resulting aggregate subgradient $\tilde{\boldsymbol{\xi}}_{k+1}$ and the aggregate subgradient locality measure $\tilde{\beta}_{k+1}$ are computed from the formulae

$$\tilde{\boldsymbol{\xi}}_{k+1} = \lambda_1^k \boldsymbol{\xi}_m + \lambda_2^k \boldsymbol{\xi}_{k+1} + \lambda_3^k \tilde{\boldsymbol{\xi}}_k \quad \text{and} \quad \tilde{\beta}_{k+1} = \lambda_2^k \beta_{k+1} + \lambda_3^k \tilde{\beta}_k.$$

The line search procedure used in the LMBM is rather similar to that used in the PBM (see Sect. 15.2.3). However, due to the simple aggregation procedure above only one trial point $\mathbf{y}_{k+1} = \mathbf{x}_k + t_R^k \mathbf{d}_k$ and a corresponding subgradient $\boldsymbol{\xi}_{k+1} \in \partial f(\mathbf{y}_{k+1})$ need to be stored.

As a stopping parameter, we use the value $w_k = -\tilde{\boldsymbol{\xi}}_k^T \mathbf{d}_k + 2\tilde{\beta}_k$ and we stop if $w_k \leq \varepsilon_s$ for some user specified $\varepsilon_s > 0$. The parameter w_k is also used during the

line search procedure to represent the desirable amount of descent (cf. v_k in the PBM).

In the LMBM both the limited memory BFGS (L-BFGS) and the limited memory SR1 (L-SR1) update formulae [9] are used in calculations of the search direction and the aggregate values. The idea of limited memory matrix updating is that instead of storing large $n \times n$ matrices D_k, one stores a certain (usually small) number of vectors obtained at the previous iterations of the algorithm, and uses these vectors to implicitly define the variable metric matrices. In the case of a null step, we use the L-SR1 update, since this update formula allows us to preserve the boundedness and some other properties of generated matrices which guarantee the global convergence of the method. Otherwise, since these properties are not required after a serious step, the more efficient L-BFGS update is employed (for more details, see [15, 16]).

Under the upper semi-smoothness assumption [6] the LMBM can be proved to be globally convergent for locally Lipschitz continuous objective functions [16].

15.2.6 Discrete Gradient Method (DGM)

Next we briefly describe the discrete gradient method (DGM) [3]. The idea of the DGM is to hybridize derivative free methods with bundle methods. That is, the DGM approximates subgradients by discrete gradients using function values only. Similarly to bundle methods, the previous values of discrete gradients are gathered into a bundle and the null step is used if the current search direction is not good enough.

We start with the definition of the discrete gradient. Let us denote by

$$S_1 = \left\{\mathbf{g} \in \mathbb{R}^n \mid \|\mathbf{g}\| = 1\right\}$$

the sphere of the unit ball and by

$$P = \left\{z \mid z : \mathbb{R}_+ \to \mathbb{R}_+,\ \lambda > 0,\ \lambda^{-1} z(\lambda) \to 0,\ \lambda \to 0\right\}$$

the set of univariate positive infinitesimal functions. In addition, let

$$G = \left\{\mathbf{e} \in \mathbb{R}^n \mid \mathbf{e} = (e_1, \ldots, e_n),\ |e_j| = 1,\ j = 1, \ldots, n\right\}$$

be a set of all vertices of the unit hypercube in $\mathbb{R}^n$. We take any $\mathbf{g} \in S_1$, $\mathbf{e} \in G$, $z \in P$, a positive number $\alpha \in (0, 1]$, and we compute $i = \arg\max\{|g_j|,\ j = 1, \ldots, n\}$. For $\mathbf{e} \in G$ we define the sequence of n vectors $\mathbf{e}^j(\alpha) = (\alpha e_1, \alpha^2 e_2, \ldots, \alpha^j e_j, 0, \ldots, 0)$ $j = 1, \ldots, n$ and for $\mathbf{x} \in \mathbb{R}^n$ and $\lambda > 0$, we consider the points

$$\mathbf{x}_0 = \mathbf{x} + \lambda \mathbf{g}, \qquad \mathbf{x}_j = \mathbf{x}_0 + z(\lambda)\mathbf{e}^j(\alpha), \quad j = 1, \ldots, n.$$

Definition 15.1 The *discrete gradient* of the function f at the point $\mathbf{x} \in \mathbb{R}^n$ is the vector $\Gamma^i(\mathbf{x}, \mathbf{g}, \mathbf{e}, z, \lambda, \alpha) = (\Gamma_1^i, \ldots, \Gamma_n^i) \in \mathbb{R}^n$ with the following coordinates:

$$\Gamma_j^i = \left[z(\lambda)\alpha^j e_j)\right]^{-1}\left[f(\mathbf{x}_j) - f(\mathbf{x}_{j-1})\right], \quad j = 1, \ldots, n,\ j \neq i,$$

$$\Gamma_i^i = (\lambda g_i)^{-1}\left[f(\mathbf{x}+\lambda\mathbf{g}) - f(\mathbf{x}) - \lambda \sum_{j=1, j\neq i}^{n} \Gamma_j^i g_j\right].$$

It has been proved in [3] that the closed convex set of discrete gradients

$$D_0(\mathbf{x},\lambda) = \mathrm{cl}\,\mathrm{conv}\big\{\mathbf{v}\in\mathbb{R}^n \mid \exists \mathbf{g}\in S_1,\ \mathbf{e}\in G,\ z\in P \text{ such that } \mathbf{v} = \Gamma^i(\mathbf{x},\mathbf{g},\mathbf{e},z,\lambda,\alpha)\big\}$$

is an approximation to the subdifferential $\partial f(\mathbf{x})$ for sufficiently small $\lambda > 0$. Thus, it can be used to compute the descent direction for the objective. However, the computation of the whole set $D_0(\mathbf{x},\lambda)$ is not easy, and therefore, in the DGM we use only a few discrete gradients from the set to calculate the descent direction.

Let us denote by l the index of the subiteration in the direction-finding procedure, by k the index of the outer iteration, and by s the index of the inner iteration. In what follows we use only the iteration counter l whenever possible without confusion. At every iteration k_s we first compute the discrete gradient $\mathbf{v}_1 = \Gamma^i(\mathbf{x},\mathbf{g}_1,\mathbf{e},z,\lambda,\alpha)$ with respect to any initial direction $\mathbf{g}_1 \in S_1$ and we set the initial bundle of discrete gradients $\bar{D}_1(\mathbf{x}) = \{\mathbf{v}_1\}$. Then we compute the vector

$$\mathbf{w}_l = \arg\min_{\mathbf{w}\in\bar{D}_l(\mathbf{x})} \|\mathbf{w}\|^2,$$

that is the distance between the convex hull $\bar{D}_l(\mathbf{x})$ of all computed discrete gradients and the origin. If this distance is less than a given tolerance $\delta > 0$, we accept the point $\mathbf{x}$ as an approximate stationary point and go to the next outer iteration. Otherwise, we compute another search direction

$$\mathbf{g}_{l+1} = -\frac{\mathbf{w}_l}{\|\mathbf{w}_l\|}$$

and we check whether this direction is descent. If it is, we have

$$f(\mathbf{x}+\lambda\mathbf{g}_{l+1}) - f(\mathbf{x}) \leq -c_1\lambda\|\mathbf{w}_l\|,$$

with the given numbers $c_1 \in (0,1)$ and $\lambda > 0$. Then we set $\mathbf{d}_{k_s} = \mathbf{g}_{l+1}$, $\mathbf{v}_{k_s} = \mathbf{w}_l$ and stop the direction finding procedure. Otherwise, we compute another discrete gradient $\mathbf{v}_{l+1} = \Gamma^i(\mathbf{x},\mathbf{g}_{l+1},\mathbf{e},z,\lambda,\alpha)$ into the direction $\mathbf{g}_{l+1}$, update the bundle of discrete gradients

$$\bar{D}_{l+1}(\mathbf{x}) = \mathrm{conv}\big\{\bar{D}_l(\mathbf{x}) \cup \{\mathbf{v}_{l+1}\}\big\}$$

and continue the direction finding procedure with $l = l+1$. Note that at each subiteration the approximation of the subdifferential $\partial f(\mathbf{x})$ is improved. It has been proved in [3] that the direction finding procedure is terminating.

In [3], it is proved that the DGM is globally convergent for locally Lipschitz continuous functions under the assumption that the set of discrete gradients uniformly approximates the subdifferential.

15.2.7 Quasisecant Method (QSM)

In this subsection, we briefly describe the quasisecant method (QSM) [2]. Here, it is again assumed that one can compute both the function value and one subgradient at any point.

The QSM can be considered as a hybrid of bundle methods and the gradient sampling method [8]. The method builds up information about the approximation of the subdifferential using a bundling idea, which makes it similar to bundle methods, while subgradients are computed from a given neighborhood of a current iteration point, which makes the method similar to the gradient sampling method.

We start this subsection with the definition of a quasisecant for locally Lipschitz continuous functions.

Definition 15.2 A vector $\mathbf{v} \in \mathbb{R}^n$ is called a *quasisecant* of the function f at the point $\mathbf{x} \in \mathbb{R}^n$ in the direction $\mathbf{g} \in S_1$ with the length $h > 0$ if

$$f(\mathbf{x} + h\mathbf{g}) - f(\mathbf{x}) \leq h\mathbf{v}^T\mathbf{g}.$$

We will denote this quasisecant by $\mathbf{v}(\mathbf{x}, \mathbf{g}, h)$.

For a given $h > 0$ let us consider the set of quasisecants at a point $\mathbf{x}$

$$QSec(\mathbf{x}, h) = \left\{\mathbf{w} \in \mathbb{R}^n \mid \exists\, \mathbf{g} \in S_1 \text{ s.t. } \mathbf{w} = \mathbf{v}(\mathbf{x}, \mathbf{g}, h)\right\}$$

and the set of limit points of quasisecants as $h \searrow 0$:

$$QSL(\mathbf{x}) = \Big\{\mathbf{w} \in \mathbb{R}^n \mid \exists \mathbf{g} \in S_1,\ h_k > 0,\ h_k \searrow 0 \text{ when } k \to \infty$$

$$\text{s.t. } \mathbf{w} = \lim_{k\to\infty} \mathbf{v}(\mathbf{x}, \mathbf{g}, h_k)\Big\}.$$

A mapping $\mathbf{x} \mapsto QSec(\mathbf{x}, h)$ is called a *subgradient-related (SR)-quasisecant mapping* if the corresponding set $QSL(\mathbf{x}) \subseteq \partial f(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$. In this case, the elements of $QSec(\mathbf{x}, h)$ are called *SR-quasisecants*. In the sequel, we will consider sets $QSec(\mathbf{x}, h)$ which contain only SR-quasisecants.

It has been shown in [2] that the closed convex set of quasisecants

$$W_0(\mathbf{x}, h) = \operatorname{cl}\operatorname{conv} QSec(\mathbf{x}, h)$$

can be used to find a descent direction for the objective with any $h > 0$. However, it is not easy to compute the entire set $W_0(\mathbf{x}, h)$, and therefore we use only a few quasisecants from the set to calculate the descent direction in the QSM.

The procedures used in the QSM are pretty similar to those in the DGM but instead of the discrete gradient $\mathbf{v}_l = \Gamma^i(\mathbf{x}, \mathbf{g}_l, \mathbf{e}, z, \lambda, \alpha)$ we use here the quasisecant $\mathbf{v}_l(\mathbf{x}, \mathbf{g}_l, h)$. Thus, at every iteration k_s we compute the vector

$$\mathbf{w}_l = \arg\min_{\mathbf{w}\in\bar{V}_l(\mathbf{x})} \|\mathbf{w}\|^2,$$

Table 15.1 Tested pieces of software

Software	Author(s)	Method	Reference
SUBG	Karmitsa	Subgradient	[42]
SolvOpt	Kuntsevich & Kappel	Shor's r-algorithm	[19, 25, 42]
PBNCGC	Mäkelä	Proximal bundle	[33, 35]
PNEW	Lukšan & Vlček	Bundle-Newton	[29]
LMBM	Karmitsa	Limited memory bundle	[15, 16]
DGM	Bagirov et al.	Discrete Gradient	[3]
QSM	Bagirov & Ganjehlou	QuasiSecant	[2]

where $\bar{V}_l(\mathbf{x})$ is a set of all quasisecants computed so far. If $\|\mathbf{w}_l\| < \delta$ with a given tolerance $\delta > 0$, we accept the point $\mathbf{x}$ as an approximate stationary point, a so-called (h, δ)-*stationary point* [2], and we go to the next outer iteration. Otherwise, we compute another search direction $\mathbf{g}_{l+1} = -\mathbf{w}_l/\|\mathbf{w}_l\|$ and we check whether this direction is descent or not. If it is, we set $\mathbf{d}_{k_s} = \mathbf{g}_{l+1}$, $\mathbf{v}_{k_s} = \mathbf{w}_l$ and stop the direction-finding procedure. Otherwise, we compute another quasisecant $\mathbf{v}_{l+1}(\mathbf{x}, \mathbf{g}_{l+1}, h)$, update the bundle of quasisecants $\bar{V}_{l+1}(\mathbf{x}) = \text{conv}\{\bar{V}_l(\mathbf{x}) \cup \{\mathbf{v}_{l+1}(\mathbf{x}, \mathbf{g}_{l+1}, h)\}\}$ and continue the direction-finding procedure with $l = l + 1$. It has been proved in [2] that the direction-finding procedure is terminating. When the descent direction $\mathbf{d}_{k_s}$ has been found, we need to compute the next (inner) iteration point similarly to that in the DGM.

The QSM is globally convergent for locally Lipschitz continuous functions under the assumption that the set $QSec(\mathbf{x}, h)$ is a SR-quasisecant mapping, that is, quasisecants can be computed using subgradients [2].

15.3 Numerical Experiments

In what follows, we compare the implementations of the methods described above. The more detailed description about the test results can be found in [21].

15.3.1 Solvers

The tested optimization codes are presented in Table 15.1. The codes or links for downloading the codes are available from http://napsu.karmitsa.fi/nsosoftware/. The experiments were performed on an Intel® Core™ 2 CPU 1.80 GHz.

SUBG is a crude implementation of the basic subgradient algorithm. The step length is chosen to be to some extent constant. We use the following three criteria as a stopping rule for SUBG: the number of function evaluations (and iterations) is restricted by a parameter and also the algorithm stops if either it cannot decrease the

value of the objective function within some successive iterations, or it cannot find a descent direction within some successive iterations. Since a subgradient method is not a descent method, we store the best value f_{best} of the objective function and the corresponding point $\mathbf{x}_{best}$ and return them as a solution if any of the stopping rules above is met.

SolvOpt (a solver for local nonlinear optimization problems) is an implementation of Shor's r-algorithm. The approaches used to handle the difficulties with step size selection and termination criteria in Shor's r-algorithm are heuristic (for details see [19]). In SolvOpt one can select to use either original subgradients or their difference approximations (i.e. the user does not have to code difference approximations but to select one parameter to do this automatically). In our experiments we have used both analytically and numerically calculated subgradients. In what follows, we denote SolvOptA and SolvOptN, respectively, the corresponding solvers. There exist MatLab, C, and Fortran source codes for SolvOpt. In our experiments we used SolvOpt v.1.1 HP-UX FORTRAN-90 sources. To compile the code, we used gfortran, the GNU Fortran 95 compiler.

PBNCGC is an implementation of the most frequently used bundle method in NSO, that is, the proximal bundle method. The code includes the constraint handling (bound constraints, linear constraints, and nonlinear/nonsmooth constraints). The quadratic direction-finding problem (15.4) is solved by the subroutine PLQDF1 implementing dual projected gradient method proposed in [27].

PNEW is a bundle-Newton solver for unconstrained and linearly constrained NSO. We used the numerical calculation of the Hessian matrix in our experiments (this can be done automatically). The quadratic direction-finding problem (15.11) is solved by the same subroutine PLQDF1 [27] like in PBNCGC.

LMBM is an implementation of a limited memory bundle method specifically developed for large-scale nonsmooth problems. In our experiments we used the adaptive version of the code with the initial number of stored correction pairs (used to form the variable metric update) equal to 7 and the maximum number of stored correction pairs equal to 15. These values have been chosen according to the numerical experiments.

DGM is a discrete gradient solver for derivative free optimization. To apply DGM, one only needs to be able to compute at every point $\mathbf{x}$ the value of the objective function and the subgradient will be approximated.

QSM is a quasisecant solver for nonsmooth, possibly nonconvex minimization. We have used both analytically calculated subgradients and approximated subgradients in our experiments (this can be done automatically by selecting one parameter). In what follows, we denote QSMA and QSMN, respectively, the corresponding solvers.

All the algorithms but SolvOpt were implemented in Fortran77 with double-precision arithmetic. To compile the codes, we used g77, the GNU Fortran 77 compiler.

15.3.2 Problems

We consider ten types of problems:

XSC: Extra-small convex problems, $n \leq 20$ ([31, Problems 2.1–2.7, 2.9, 2.22 and 2.23, and 3.4–3.8, 3.10, 3.12, 3.16, 3.19 and 3.20]);
XSNC: Extra-small nonconvex problems ([31, Problems 2.8, 2.10–2.12, 2.14–2.16, 2.18–2.21, 2.24 and 2.25, and 3.1, 3.2, 3.15, 3.17, 3.18 and 3.25]);
SC: Small-scale convex problems, $n = 50$ ([15, Problems 1–5], Problems 2 and 5 in `TEST29` [28], and six maximum of quadratic functions [21]);
SNC: Small-scale nonconvex problems ([15, Problems 6–10], and Problems 13, 17 and 22 in `TEST29` [28], and six maximum of quadratic functions);
MC and MNC: Medium-scale convex and nonconvex problems, $n = 200$ (see SC and SNC problems);
LC and LNC: Large-scale convex and nonconvex problems, $n = 1000$ (see MC and MNC problems);
XLC and XLNC: Extra-large-scale convex and nonconvex problems, $n = 4000$ (see MC and MNC problems but only two maximum of quadratics with a diagonal matrix);

Problems 2, 5, 13, 17, and 22 in `TEST29` are from the software package UFO (Universal Functional Optimization) [28]. The problems were selected so that in all cases all the solvers converged to the same local minimum. However, it is worth mentioning that, in the case of different local minima (i.e. in some nonconvex problems omitted from the test set), solvers `LMBM`, `SolvOpt`, and `SUBG` usually converged to the same local minimum, while `PBNCGC`, `DGM`, and `QSM` converged to a different local minimum. The solver `PNEW` converged sometimes with the first group and some other times with the second. Moreover, `DGM` and `QSM` seem to have an aptitude for finding global or at least smaller local minima than the other solvers. For example, in Problems 3.13 and 3.14 in [31] all the other solvers converged to the minimum reported in [31] but `DGM` and `QSM` "converged" to minus infinity.

15.3.3 Termination, Parameters, and Acceptance of the Results

The determination of stopping criteria for different solvers, such that the comparison of different methods is fair, is not a trivial task.

We say that a solver finds the solution with respect to a tolerance $\varepsilon > 0$ if

$$\frac{f_{best} - f_{opt}}{1 + |f_{opt}|} \leq \varepsilon,$$

where f_{best} is a solution obtained with the solver and f_{opt} is the best-known (or optimal) solution.

We fixed the stopping criteria and parameters for the solvers using three different problems from three different problem classes: Problem 2.4 in [31] (XSC), Problem 3.15 in [31] (XSNC), and Problem 3 in [15] with $n = 50$ (SC). With all the solvers we sought the loosest termination parameters such that the results for all the three test problems were still acceptable with respect to the tolerance $\varepsilon = 10^{-4}$. In addition to the usual stopping criteria of the solvers, we terminated the experiments if the elapsed CPU time exceeded half an hour.

We have accepted the results for XS and S problems ($n \leq 50$) with respect to the tolerance $\varepsilon = 5 \times 10^{-4}$. With larger problems ($n \geq 200$), we have accepted the results with the tolerance $\varepsilon = 10^{-3}$. In what follows, we report also the results for all problem classes with respect to the relaxed tolerance $\varepsilon = 10^{-2}$ to have an insight into the reliability of the solvers (i.e. is a failure a real failure or is it just an inaccurate result which could possible be prevented with a more tight stopping parameter).

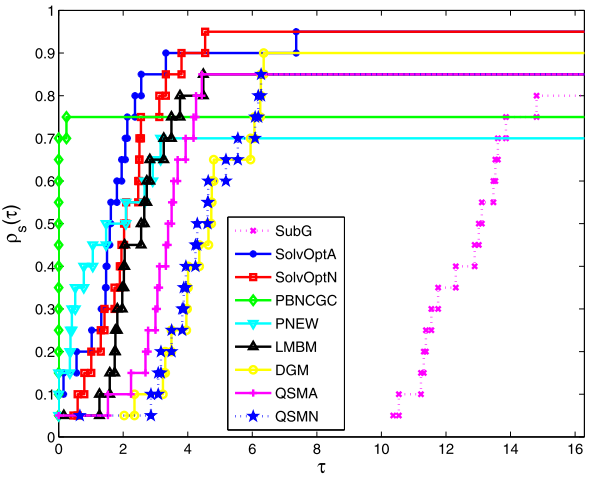
With all the bundle-based solvers the distance measure parameter value $\gamma = 0.5$ was used with nonconvex problems. With `PBNCGC` and `LMBM` the value $\gamma = 0$ was used with convex problems and, since with `PNEW` γ has to be positive, $\gamma = 10^{-10}$ was used with `PNEW`. For those solvers storing subgradients (or approximations of subgradients)—that is, `PBNCGC`, `PNEW`, `LMBM`, `DGM`, and `QSM`—the maximum size of the bundle was set to $\min\{n+3, 100\}$. For all other parameters we used the default settings of the codes.

15.3.4 Results

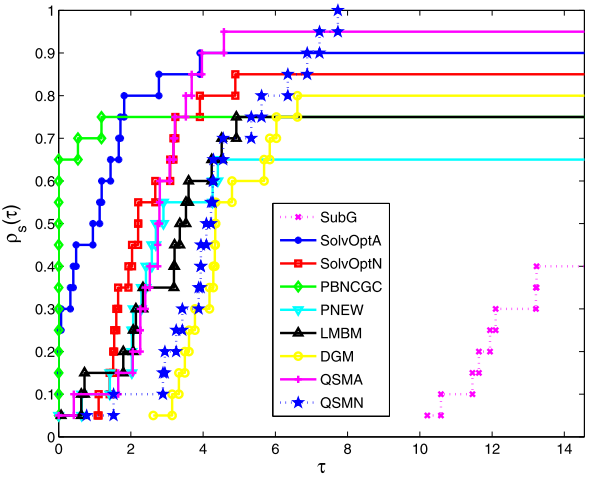
The results are summarized in Figs. 15.1–15.8 and in Table 15.2. The results are analyzed using the performance profiles introduced in [13]. We compare the efficiency of the solvers both in terms of computational times and numbers of function and subgradient evaluations (evaluations for short). In the performance profiles, the value of $\rho_s(\tau)$ at $\tau = 0$ gives the percentage of test problems for which the corresponding solver is the best (it uses least computational time or evaluations) and the value of $\rho_s(\tau)$ at the rightmost abscissa gives the percentage of test problems that the corresponding solver can solve, that is, the reliability of the solver (this does not depend on the measured performance). Moreover, the relative efficiency of each solver can be directly seen from the performance profiles: the higher the particular curve, the better the corresponding solver. For more information on performance profiles, see [13].

15.3.4.1 Extra-Small Problems

There was not a big difference in the computational times of the different solvers when solving the XS problems. Thus, only the numbers of function and subgradient evaluations are reported in Fig. 15.1.



(a) Convex



(b) Nonconvex

Fig. 15.1 Evaluations for XS problems (20 problems with $n \leq 20$, $\varepsilon = 5 \times 10^{-4}$)

PBNCGC was usually the most efficient solver when comparing the numbers of evaluations. This is, in fact, true for all sizes of problems. Thus, PBNCGC should be a good choice as a solver in case the objective function value and/or the subgradient are expensive to compute. However, PBNCGC failed to achieve the desired accuracy in 25 % of the extra-small problems (both XSC and XSNC) which means that it had almost the worst degree of success in solving these problems.

SUBG is highly unsuitable for nonconvex problems: it failed in 60 % of the problems ($\varepsilon = 5 \times 10^{-4}$, see Fig. 15.1(b)). On the other hand, SolvOpt was one of the most reliable solvers together with QSM in both convex and nonconvex settings although, theoretically, Shor's r-algorithm is not supposed to solve noncon-

vex problems. SolvOptA was also the most efficient method except for PBNCGC (especially in the nonconvex case).

Except for SUBG, the solvers did not have big differences in the numbers of success in solving XSC or XSNC problems. However, it is noteworthy that the QSM computed nonconvex problems more reliably than convex ones.

Most of the failures reported here are, in fact, inaccurate results: all the solvers but PNEW succeed in solving equal or more than 95 % of XSC problems with respect to the relaxed tolerance $\varepsilon = 10^{-2}$. The corresponding percentage for XSNC problems was 85 %, although here also SUBG failed to solve so many problems.

In XSC problems PNEW was the second most efficient solver (see Fig. 15.1(a)). However, it failed to solve 30 % of the convex problems and 35 % of the nonconvex problems. The reason for this relatively large number of failures with PNEW is in its sensitivity to the internal parameter XMAX (RPAR(9) in the code) which is noted also in [30]. If we, instead of only one (default) value, used a selected value for this parameter, also the solver PNEW solved 85 % of XSNC problems.

The derivative-free solvers DGM and QSMN performed similarly in these small-scale problems but QSMN was clearly more reliable in the nonconvex case. SolvOptN usually used less evaluations than the derivative-free solvers both in XSC and XSNC problems. However, in the nonconvex case, also SolvOptN lost out to QSMN in reliability.

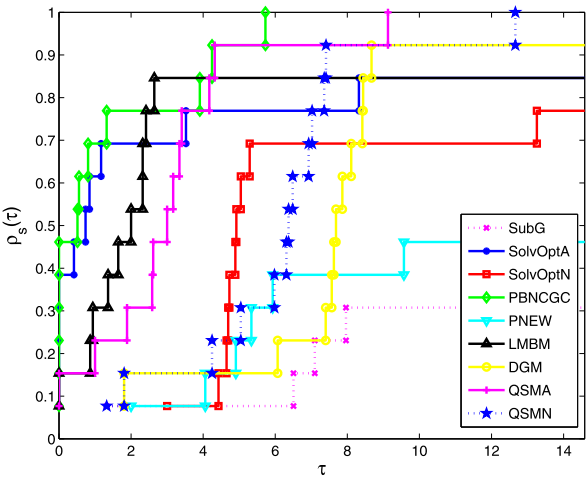
15.3.4.2 Small-Scale Problems

Already with small-scale problems, there was a wide diversity on the computational times of different codes. Moreover, the numbers of evaluations used with solvers were no longer directly comparable with the elapsed computational times. For instance, PBNCGC was clearly the winner when comparing the numbers of evaluations (see Figs. 15.2(b) and 15.3(b)). However, when comparing computational times, SolvOptA was equally efficient with PBNCGC in SC problems (see Fig. 15.2(a)) and LMBM was the most efficient solver in SNC problems (see Fig. 15.3(a)).

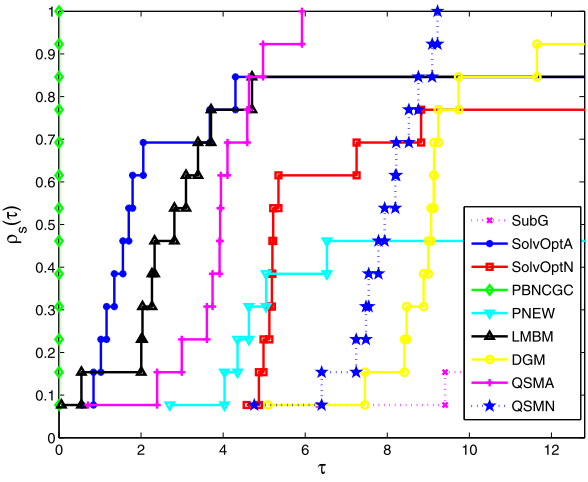
SUBG was clearly the worst solver with respect to both computational times and evaluations in both SC and SNC problems. It was also the most unreliable solver. It solved only about 30 % of the convex and 20 % of the nonconvex problems and it failed in all the quadratic problems.

Also the other subgradient solver SolvOpt had some difficulties with the accuracy, especially in the nonconvex case. SolvOptN solved about 77 % of the convex problems with respect to tolerance $\varepsilon = 5 \times 10^{-4}$ and 92 % with $\varepsilon = 10^{-2}$. For SolvOptA the corresponding values were 85 % and 92 %. In the nonconvex case, the values were 64 % vs. 92 % for SolvOptN and 71 % vs. 86 % for SolvOptA. In other words, SolvOpt would have benefited most if we instead of tolerance $\varepsilon = 5 \times 10^{-4}$ had used the relaxed tolerance $\varepsilon = 10^{-2}$ to accept the results. Note, however, that with small-scale problems SolvOpt was one of the most reliable solvers.

With the other solvers there were no big differences in solving convex or nonconvex problems apart from PNEW: PNEW solved about 79 % of the nonconvex



(a) CPU-time

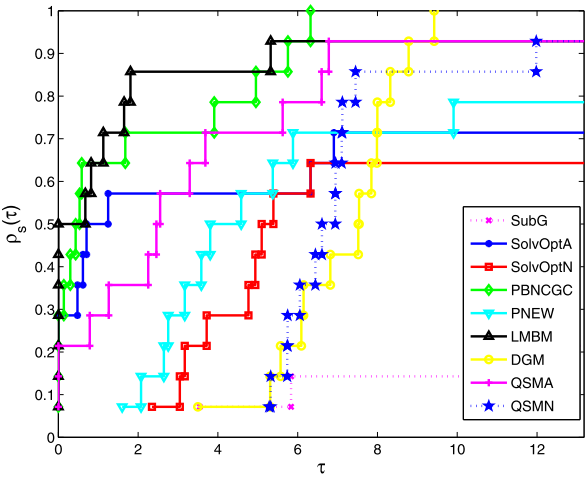


(b) Evaluations

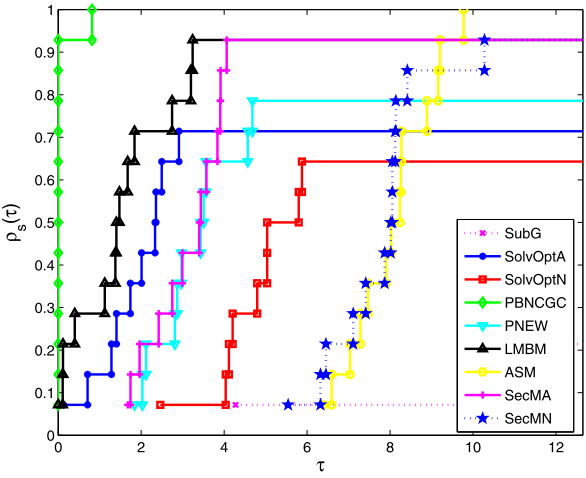
Fig. 15.2 CPU-time and evaluations for SC problems (13 problems with $n = 50$, $\varepsilon = 5 \times 10^{-4}$)

problems and only 46 % of the convex problems. Also LMBM succeeded in solving a little bit more nonconvex than convex problems. In the convex case, PBNCGC, QSMA, and QSMN succeeded in solving all the problems with the desired accuracy. With the relaxed tolerance $\varepsilon = 10^{-2}$ also DGM managed to solve all the problems and all the solvers but PNEW and SUBG succeeded in solving more than 90 % of the problems. In the nonconvex case, PBNCGC and DGM solved all the problems successfully. With a relaxed parameter QSMA and QSMN succeeded as well and all the solvers except PNEW and SUBG managed to solve more than 85 % of the problems.

The solvers DGM and QSMN behaved rather similarly but QSMN was a little bit more efficient both with respect to computational times and evaluations. SolvOptN outperformed these two methods in efficiency but lost clearly in reliability.



(a) CPU-time

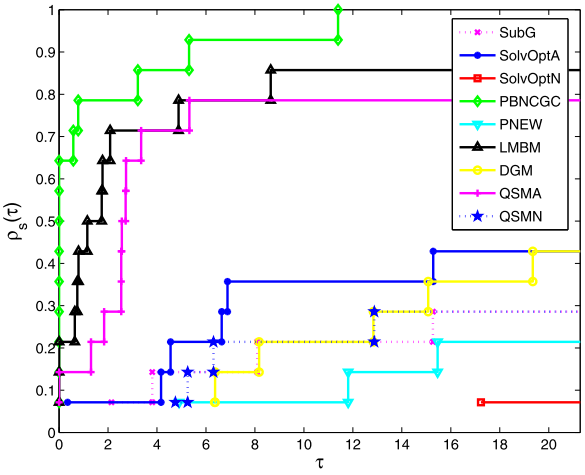


(b) Evaluations

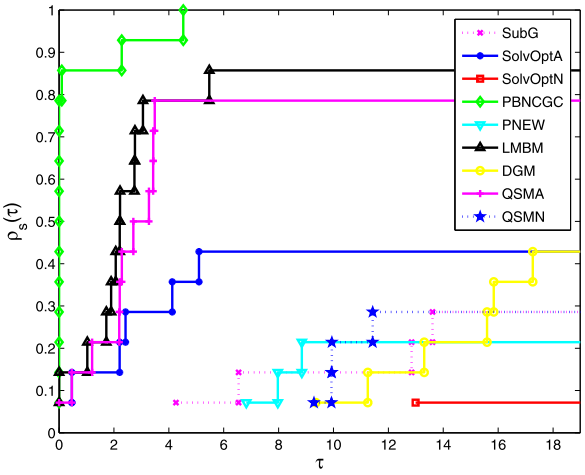
Fig. 15.3 CPU-time and evaluations for SNC problems (14 problems with $n = 50$, $\varepsilon = 5 \times 10^{-4}$)

PNEW failed to solve all but one of the convex quadratic problems and succeeded in solving all but one non-quadratic problems. In the nonconvex case PNEW succeeded in solving all the quadratic problems but then it had some difficulties with the other problems. Again, the reason for these failures is in its sensitivity to the internal parameter XMAX.

In [34], PNEW is reported to be very efficient in quadratic problems. Also in our experiments, PNEW was clearly more efficient with the quadratic problems than with the non-quadratic. However, except for some small problems, it was not the most efficient method in any of the cases.



(a) CPU-time



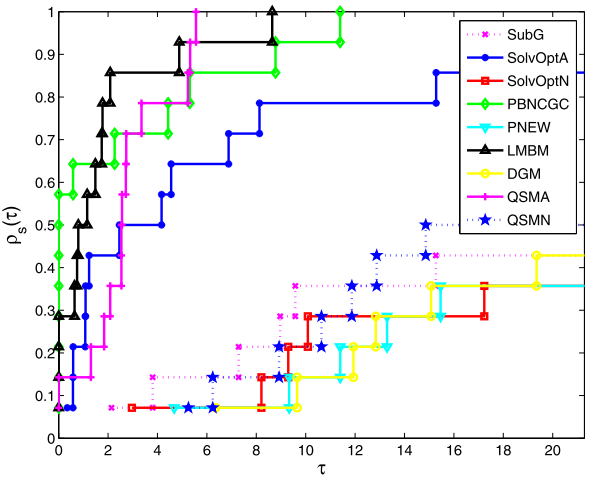
(b) Evaluations

Fig. 15.4 CPU-time and evaluations for LNC problems (14 problems with $n = 1000$, $\varepsilon = 10^{-3}$)

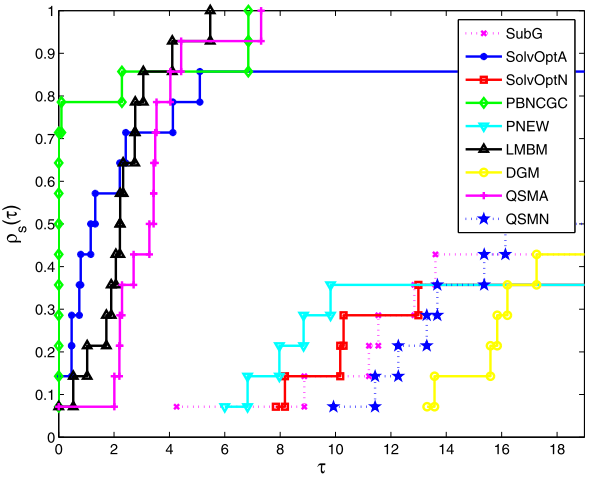
15.3.4.3 Medium and Large-Scale Problems

The results for medium and large-scale problems reveal similar trends. Thus, we show here only the results for large problems in Figs. 15.4 and 15.5. More illustrated results also for medium-scale problems can be found in [21].

When solving medium and large-scale problems, the solvers are divided into two groups: the first group consists of more efficient solvers: `LMBM`, `PBNCGC`, `QSMA`, and `SolvOptA`. The second group consists of solvers using some kind of approximation for subgradients or Hessian, and `SUBG`. In the nonconvex case (see Fig. 15.4), the inaccuracy of `SolvOptA` made it slide to the group of less efficient solvers. In Fig. 15.5 illustrating the results with the relaxed tolerance, `SolvOptA`



(a) CPU-time



(b) Evaluations

Fig. 15.5 CPU-time and evaluations for LNC problems, low accuracy (14 problems with $n = 1000$, $\varepsilon = 10^{-2}$)

is among the more efficient solvers. Nevertheless, its accuracy is not as good as that of the others. At the same time, the successfully solved quadratic problems almost lifted `PNEW` to the group of more efficient solvers in large-scale nonconvex settings (especially when comparing the numbers of evaluations, see [21]). The similar trend cannot have been seen here, since in LNC problems the time limit was exceeded in all the maximum of quadratic problems with `PNEW`.

Although `PBNCGC` was usually (on 70 % of medium-sized and 60 % of large problems) the most efficient solver tested in the convex case, it was also the one that needed the longest time to compute Problem 3 in [15] (both in medium and large-scale settings). Indeed, an average time used to solve an MC (LC) problem with `PBNCGC` was 15.7 (266.0) seconds while with `SolvOptA`, `LMBM`, and `QSMA` they

were 1.3 (22.0), 1.6 (54.5), and 4.9 (98.6) seconds, respectively (the average times are calculated using 9 (7) problems that all the solvers above managed to solve).

In the MNC case, LMBM and PBNCGC were the most efficient solvers. However, also here with PBNCGC there was notable variation in the computational times for different problems while with LMBM all the problems were solved equally efficiently. In LNC settings also the solver QSMA solved the problems quite efficiently (see Figs. 15.4 and 15.5).

The efficiency of PBNCGC is mostly due to its efficiency in quadratic problems: it was the most efficient solver in almost all quadratic problems when comparing the computational times, and superior when comparing the numbers of evaluations. As before, PNEW failed in all but one of the convex quadratic problems.

Besides usually being the most efficient solver, PBNCGC was also the most reliable solver tested in medium-scale settings. In the MC case it was the only solver that succeeded in solving all the problems with the desired accuracy. In the MNC case QSMA was successful as well. With the relaxed tolerance $\varepsilon = 10^{-2}$ also SolvOptA, QSMA, QSMN, and DGM managed to solve all the MC problems, while LMBM and SolvOptN succeeded in solving more than 84 % of the problems. In the MNC case, LMBM, PBNCGC QSMA, QSMN, and DGM solved all the problems with the relaxed tolerance.

SolvOptN had some serious difficulties with the accuracy, especially in nonconvex cases. For instance, with the relaxed tolerance SolvOptN solved almost 80 % of the MNC problems while with the tolerance $\varepsilon = 10^{-3}$ less than 30 %. A similar effect could be seen with SolvOptA, although not as pronounced. Naturally, with the LNC problems the difficulties with the accuracy degenerated (see Figs. 15.4 and 15.5).

Also LMBM and QSMA had some difficulties with the accuracy in the LNC case (see Fig. 15.4). With the relaxed tolerance, they solved all LNC problems (see Fig. 15.5). With this tolerance LMBM was clearly the most efficient solver in nonquadratic problems and the computational times of both LMBM and QSMA were comparable with those of PBNCGC in the whole test set.

The solvers PBNCGC, DGM, and QSM were the only solvers which solved two LC problems in which there is only one nonzero element in the subgradient vector (i.e. Problem 1 in [15] and Problem 2 in TEST29 [28]). With the other methods, there were some difficulties already with $n = 50$ and some more with $n = 200$. (Note that for small, medium and large-scale settings, the problems are the same, only the number of variables is changing.) In the case of LMBM these difficulties are easy to explain: the approximation of the Hessian formed during the calculations is dense and, naturally, not even close to the real Hessian in sparse problems. It has been reported [15] that LMBM is best suited for the problems with a dense subgradient vector whose components depend on the current iteration point. This result is in line with the noted result that LMBM solves nonconvex problems very efficiently.

In the LC case PNEW solved all but the above mentioned two problems and the maximum of quadratics problems. The solvers DGM, LMBM, SUBG, and QSMN failed to solve (possible in addition to the two above-mentioned problems) two piecewise linear problems (Problem 2 in [15] and Problem 5 in TEST29 [28]) and QSMA also failed to solve one of them.

Naturally, for the solvers using difference approximation or some other approximation based on the calculation of the function or subgradient values, the number of evaluations (and thus also the computational time) grows enormously when the number of variables increases. Particularly, in large-scale problems the time limit was exceeded with all these solvers in all the maximum of quadratic problems. Thus, the number of failures with these solvers is probably larger than it should be. Nevertheless, if you need to solve a problem where the subgradient is not available, the best solver would probably be `SolvOptN` (only in the convex case) due to its efficiency or `QSMN` due to its reliability.

15.3.4.4 Extra-Large Problems

Finally, we tested the most efficient solvers so far, that is `LMBM`, `PBNCGC`, `QSMA` and `SolvOptA`, using the problems with $n = 4000$. In the convex case, the solver `QSMA`, which has kept a rather low profile until now, was clearly the most efficient method although `PBNCGC` still used the least evaluations. `QSMA` was also the most reliable of the solvers tested (see Fig. 15.6(a)).

In the nonconvex case, `LMBM` and `QSMA` were approximately equally good in computational times, evaluations, and reliability (see Fig. 15.6(b)). Here `PBNCGC` was the most reliable solver, although with the tolerance $\varepsilon = 10^{-2}$ `QSMA` was the only solver that solved all the nonconvex problems. `LMBM` and `PBNCGC` failed in one and `SolvOpt` in two problems.

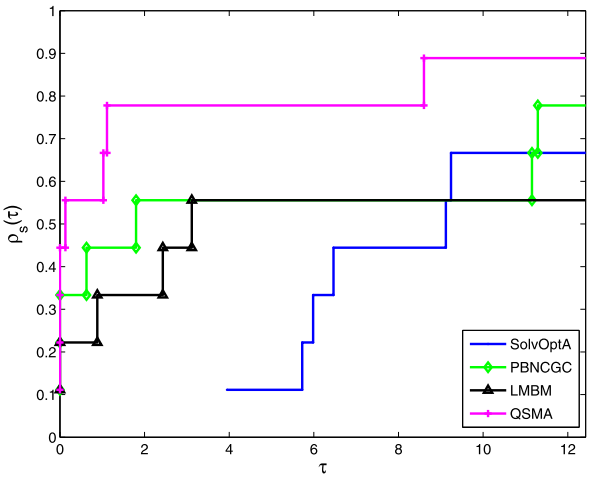
As before, `LMBM` solved all the problems it could solve in a relatively short time while with all the other solvers there was notable variation in the computational times elapsed for different problems. However, in the convex case, the efficiency of `LMBM` was again ruined by its unreliability.

15.3.4.5 Convergence Speed and Number of Success

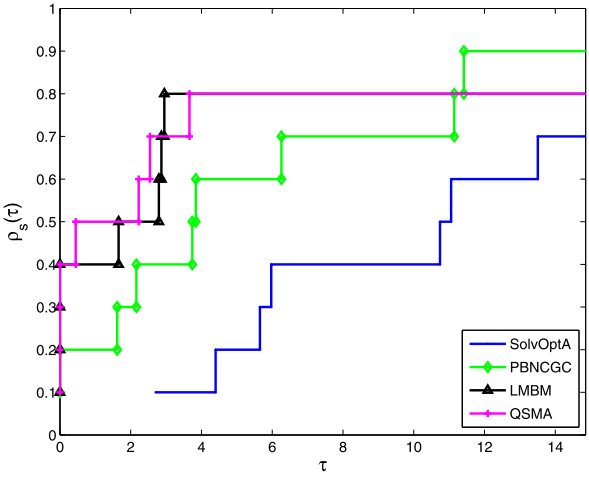
In this subsection, we first study (experimentally) the convergence speed of the algorithms using one small-scale convex problem (Problem 3 in [15]). The exact minimum value for this function (with $n = 50$) is $-49 \times 2^{1/2} \approx -69.296$.

For the limited memory bundle method the rate of convergence has not been studied theoretically. However, at least in this particular problem, the solvers `LMBM` and `PBNCGC` converged at approximately the same rate. Moreover, if we study the number of evaluations, `PBNCGC` and `LMBM` seem to have the fastest converge speed of the solvers tested (see Fig. 15.7(b)) although, theoretically, the proximal bundle method is only linearly convergent.

`SUBG` converged linearly but extremely slowly and `PNEW`, although it finally found the minimum, did not decrease the value of the function in the first 200 evaluations. Naturally, with `PNEW` a large amount of subgradient evaluations are needed to compute the approximative Hessian. The solvers `SolvOptA`, `SolvOptN`, `DGM`, `QSMA`, and `QSMN` took a very big step downwards already in iteration two (see



(a) Convex

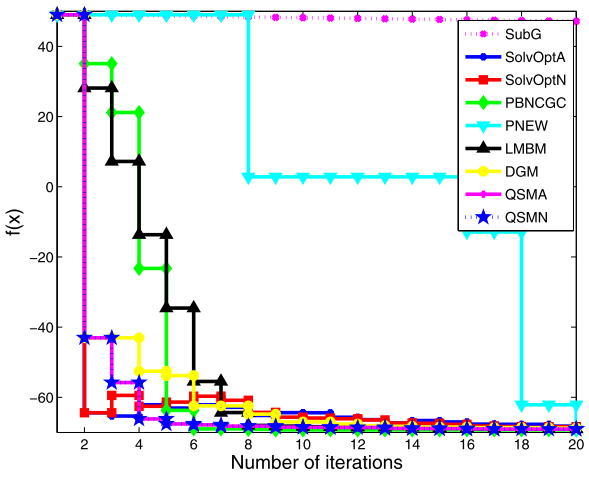


(b) Nonconvex

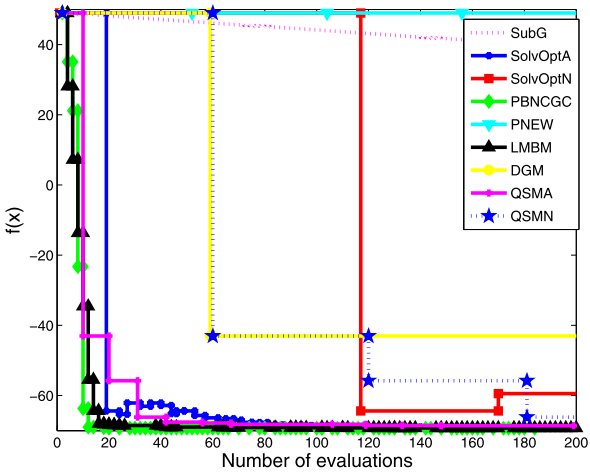
Fig. 15.6 CPU-times for convex (9 pc.) and nonconvex (10 ps.) XL problems ($n = 4000$, $\varepsilon = 10^{-3}$)

Fig. 15.7(a)). However, they took quite many function evaluations per iteration. In Fig. 15.7 it is easy to see that Shor's r-algorithm (i.e. solvers `SolvOptA` and `SolvOptN`) is not a descent method.

In order to see how quickly the solvers reach some specific level, we studied the value of the function equal to -69. With `PBNCGC` it took only 8 iterations to go below that level. The corresponding values for other solvers were 17 with `QSMA` and `QSMN`, 20 with `LMBM` and `PNEW`, and more than 20 with all the other solvers. In terms of function and subgradient evaluations, the values were 18 with `PBNCGC`, 64 with `LMBM`, and 133 with `SolvOptA`. Other solvers needed more than 200 evaluations to go below -69.



(a) First 20 iterations



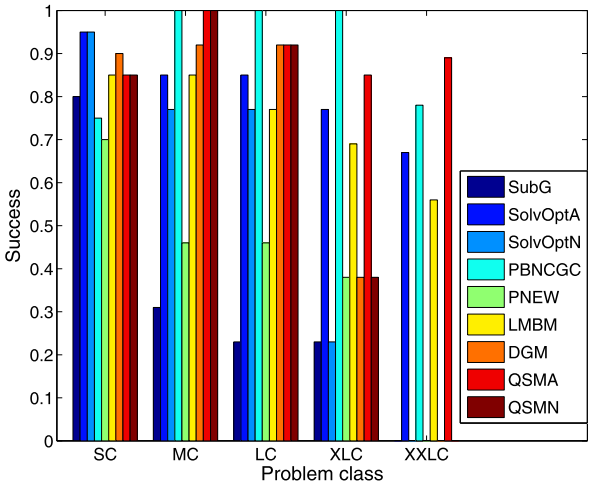
(b) First 200 evaluations

Fig. 15.7 Function values versus iterations (**a**), and function values versus the number of function and subgradient evaluations (**b**)

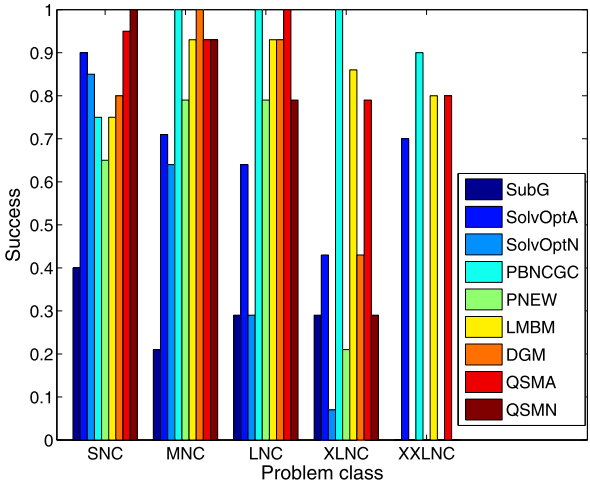
The worst of the solvers were `SUBG` which took 7382 iterations and 14764 evaluations to reach the desired accuracy and stop, and `SolvOptN` which never reached the desired accuracy (the final value obtained after 42 iterations and 2342 evaluations was -68.915).

Finally, in Fig. 15.8 we give the proportions of the successfully terminated runs obtained with each solver within the different problem classes. Although we have already said something about the reliability of the solvers, we study the figure to see if the convexity or the number of variables have any significant effect on the success rate of the solvers.

In Fig. 15.8, we see that with both variants of `SolvOpt` the degree of success decreases clearly when the number of variables increases or the problem is noncon-



(a) Convex



(b) Nonconvex

Fig. 15.8 Proportions of successfully terminated runs within different problem classes: convex problems (**a**) and nonconvex problems (**b**)

vex. In addition, with the solvers that use approximations to subgradient or Hessian there is a clear drop-out when moving from 200 variables to 1000 variables. At least one reason for this is that with $n = 1000$ the solvers terminated because of the maximum time limit (thus failing to reach the desired accuracy).

`DGM` and `QSMN` were reliable methods both with convex and nonconvex problems up to 200 variables, while `LMBM` and `PNEW` solved the nonconvex problems more reliably than the convex ones. With `PNEW` the maximum time limit was exceeded in many cases with $n = 1000$, thus the exception. With `PNEW` the result could be different if the tuned parameter `XMAX` was used. With `LMBM` the result is in harmony with the earlier claims [15] that `LMBM` works better for more nonlinear functions.

PBNCGC solved small-scale and larger problems in a very reliable way but it was almost the worst solver in extra-small problems. This result has probably nothing to do with the problem's size but more with the different problem classes used.

15.4 Conclusions

We have tested the performance of different nonsmooth optimization solvers in the solution of different nonsmooth problems. The results are summarized in Table 15.2, where we give our recommendations for the "best" solver for different problem classes. Since it is not always unambiguous what the best option is, we give credentials both in the cases where the most efficient (in terms of used computer time) and the most reliable solver are sought out. If there is more than one solver recommended in Table 15.2, the solvers are given in alphabetical order. The parenthesis in the table mean that the solver is not exactly as good as the first one but still a solver to be reckoned with the problem class.

Although in our experiments we got extremely good results with the proximal bundle solver PBNCGC, we cannot say that it is clearly the best method tested. The inaccuracy in extra-small problems, great variations in the computational times occurred in larger problems, and the earlier results obtained make us believe that our test set favored this solver over the others a little bit. Even so, we can say that PBNCGC was one of the best solvers tested and it is especially efficient for the maximum of quadratic and piecewise linear problems.

On the other hand, the limited memory bundle solver LMBM suffered from ill-conditioned test problems in convex small, medium, large and extra-large cases. In the test set there were four problems (out of 13) in which LMBM was known to have difficulties. In addition, LMBM did not beat PBNCGC in any maximum of quadratics problems but in one with $n = 4000$. This, however, is not the inferiority of LMBM but rather the superiority of PBNCGC in these kinds of problems. LMBM was quite reliable in the nonconvex case in all numbers of variables tested and it solved all the problems—even the largest ones—a in relatively short time while, for example, with PBNCGC there was great variation on the computational times of different problems. LMBM works best for (highly) nonlinear functions while for piecewise linear functions it might be a good idea to find another solver.

In convex extra-small problems, the bundle-Newton solver PNEW was the second most efficient solver tested. However, PNEW suffers greatly from the fact that it is very sensitive to the internal parameter XMAX. Already using two values for this parameter (e.g., default value 1000 and the smallest recommended value 2), the results would have been much better and especially the degree of success would have been much higher. The solver has been reported to be best suited for quadratic problems [34] and, indeed, it solved (nonconvex) quadratic problems faster that nonquadratic. However, with $n \geq 50$ it did not beat the other solvers in these problems due to the large approximation of the Hessian matrix required.

Table 15.2 Summation of the results

Problem's type	Problem's size	Seeking for efficiency	Seeking for reliability
Convex	XS	PBNCGC, PNEW[1], (SolvOpt(A+N))	DGM, SolvOpt(A+N)
	S, M, L	LMBM[2], PBNCGC, (QSMA, SolvOptA)	PBNCGC, QSMA
	XL	LMBM[2], QSMA	QSMA, (PBNCGC)
Nonconvex	XS	PBNCGC, SolvOptA, (QSMA)	QSM(A+N), (SolvOptA)
	S, M, L	LMBM, PBNCGC, (QSMA)	DGM, LMBM, PBNCGC
	XL	LMBM, QSMA	PBNCGC, (LMBM, QSMA)
Piecewise linear or sparse	XS, S	PBNCGC, SolvOptA	PBNCGC, SolvOptA
	M, L, XL	PBNCGC, QSMA[3]	DGM, PBNCGC, QSMA
Piecewise quadratic	XS	PBNCGC, PNEW[1], (LMBM, SolvOptA)	LMBM, PBNCGC, PNEW[1], SolvOptA
	S, M, L, XL	LMBM, PBNCGC, (SolvOptA)	DGM, LMBM, PBNCGC, QSMA
Highly nonlinear	XS	LMBM, PBNCGC, SolvOptA	LMBM, QSMA, SolvOptA
	S	LMBM, PBNCGC	LMBM, PBNCGC, QSMA
	M, L, XL	LMBM	LMBM, QSMA
Function evaluations are expensive	XS	PBNCGC, (PNEW[1], SolvOptA)	QSMA, SolvOptA
	S, M, L, XL	PBNCGC, (LMBM[4], SolvOptA)	PBNCGC, (LMBM[4], QSMA)
Subgradients are not available	XS	SolvOptN	QSMN, SolvOptN[5], (DGM)
	S, M	SolvOptN, QSMN	DGM, QSMN
	L	QSMN, (DGM)	DGM, QSMN

The standard subgradient solver SUBG is usable only for extra-small convex problems: the degree of success was 80 % in XSC, otherwise it was less than 40 %. In addition, the implementations of Shor's r-algorithm SolvOptA and SolvOptN did their best in extra-small problems (also in the nonconvex case!). Nevertheless, SolvOptA solved also medium, large and even extra-large problems (convex) rather efficiently. In larger nonconvex problems these methods suffered from inaccuracy.

Thus, when comparing the reliability in medium-scale settings, it seems that one should select PBNCGC for convex problems while LMBM is good for nonconvex problems. On the other hand, the quasi-secant solver QSMA was reliable and efficient both in convex and nonconvex medium-sized problems. However, with QSMA there was some variation on the computational times of different problems (not as much as PBNCGC, though) while LMBM solved all the problems in a relatively short time.

The solvers using discrete gradients, that is, the discrete gradient solver DGM and the quasisecant solver with discrete gradients, QSMN, usually lost out in efficiency to the solvers using analytical subgradients. However, in extra-small and small-scale problems the differences were not significant and the reliability of DGM and QSMN seems to be very good both with convex and nonconvex problems. Moreover in the case of highly nonconvex functions (supposing that you seek for global optimum) DGM or QSM (either with or without subgradients) would be a good choice, since these methods tend to jump over the narrow local minima.

Acknowledgements We would like to acknowledge professors A. Kuntsevich and F. Kappel for providing Shor's r-algorithm in their web-page as well as professors L. Lukšan and J. Vlček for providing the bundle-Newton algorithm. The work was financially supported by the University of Turku (Finland) and the University of Ballarat (Australia) and the Australian Research Council.

References

1. Äyrämö S (2006) Knowledge mining using robust clustering. PhD thesis, University of Jyväskylä
2. Bagirov AM, Ganjehlou AN (2010) A quasisecant method for minimizing nonsmooth functions. Optim Methods Softw 25(1):3–18
3. Bagirov AM, Karasözen B, Sezer M (2008) Discrete gradient method: Derivative-free method for nonsmooth optimization. J Optim Theory Appl 137(2):317–334
4. Beck A, Teboulle M (2003) Mirror descent and nonlinear projected subgradient methods for convex optimization. Oper Res Lett 31(3):167–175
5. Ben-Tal A, Nemirovski A (2005) Non-Euclidean restricted memory level method for large-scale convex optimization. Math Program 102(3):407–456
6. Bihain A (1984) Optimization of upper semidifferentiable functions. J Optim Theory Appl 44(4):545–568
7. Bradley PS, Fayyad UM, Mangasarian OL (1999) Mathematical programming for data mining: Formulations and challenges. INFORMS J Comput 11(3):217–238
8. Burke JV, Lewis AS, Overton ML (2005) A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM J Optim 15(3):751–779
9. Byrd RH, Nocedal J, Schnabel RB (1994) Representations of quasi-Newton matrices and their use in limited memory methods. Math Program 63(2):129–156
10. Clarke FH (1983) Optimization and nonsmooth analysis. Wiley, New York
11. Clarke FH, Ledyaev YS, Stern RJ, Wolenski PR (1998) Nonsmooth analysis and control theory. Springer, New York
12. Demyanov VF, Bagirov AM, Rubinov AM (2002) A method of truncated codifferential with application to some problems of cluster analysis. J Glob Optim 23(1):63–80
13. Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. Math Program 91(2):201–213
14. Gaudioso M, Monaco MF (1992) Variants to the cutting plane approach for convex nondifferentiable optimization. Optimization 25(1):65–75
15. Haarala M, Miettinen K, Mäkelä MM (2004) New limited memory bundle method for large-scale nonsmooth optimization. Optim Methods Softw 19(6):673–692
16. Haarala N, Miettinen K, Mäkelä MM (2007) Globally convergent limited memory bundle method for large-scale nonsmooth optimization. Math Program 109(1):181–205
17. Haslinger J, Neittaanmäki P (1996) Finite element approximation for optimal shape, material and topology design, 2nd edn. Wiley, Chichester

18. Hiriart-Urruty J-B, Lemaréchal C (1993) Convex analysis and minimization algorithms. II. Advanced theory and bundle methods. Springer, Berlin
19. Kappel F, Kuntsevich AV (2000) An implementation of Shor's r-algorithm. Comput Optim Appl 15(2):193–205
20. Kärkkäinen T, Heikkola E (2004) Robust formulations for training multilayer perceptrons. Neural Comput 16(4):837–862
21. Karmitsa N, Bagirov A, Mäkelä MM (2009) Empirical and theoretical comparisons of several nonsmooth minimization methods and software. TUCS Technical Report 959, Turku Centre for Computer Science, Turku. Available online http://tucs.fi/publications/insight.php?id=tKaBaMa09a
22. Karmitsa N, Bagirov A, Mäkelä MM (2012) Comparing different nonsmooth minimization methods and software. Optim Methods Softw 27(1):131–153. doi:10.1080/10556788.2010.526116
23. Kiwiel KC (1985) Methods of descent for nondifferentiable optimization. Lecture notes in mathematics, vol 1133. Springer, Berlin
24. Kiwiel KC (1990) Proximity control in bundle methods for convex nondifferentiable minimization. Math Program 46(1):105–122
25. Kuntsevich A, Kappel F (1997) SolvOpt – the solver for local nonlinear optimization problems. Graz. http://www.uni-graz.at/imawww/kuntsevich/solvopt/
26. Lemaréchal C (1989) Nondifferentiable optimization. In: Nemhauser GL, Rinnooy Kan AHG, Todd MJ (eds) Optimization. North-Holland, Amsterdam, pp 529–572
27. Lukšan L (1984) Dual method for solving a special problem of quadratic programming as a subproblem at linearly constrained nonlinear minimax approximation. Kybernetika 20(6):445–457
28. Lukšan L, Tůma M, Šiška M, Vlček J, Ramešová N (2002) Ufo 2002: Interactive system for universal functional optimization. Technical report V-883, Academy of Sciences of the Czech Republic, Prague
29. Lukšan L, Vlček J (1998) A bundle-Newton method for nonsmooth unconstrained minimization. Math Program 83(3):373–391
30. Lukšan L, Vlček J (2000) NDA: Algorithms for nondifferentiable optimization. Technical report V-797, Academy of Sciences of the Czech Republic, Prague
31. Lukšan L, Vlček J (2000) Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical report V-798, Academy of Sciences of the Czech Republic, Prague
32. Mäkelä MM (2002) Survey of bundle methods for nonsmooth optimization. Optim Methods Softw 17(1):1–29
33. Mäkelä MM (2003) Multiobjective proximal bundle method for nonconvex nonsmooth optimization: Fortran subroutine MPBNGC 2.0. Reports of the Department of Mathematical Information Technology, Series B, Scientific Computing B13/2003, University of Jyväskylä, Jyväskylä
34. Mäkelä MM, Miettinen M, Lukšan L, Vlček J (1999) Comparing nonsmooth nonconvex bundle methods in solving hemivariational inequalities. J Glob Optim 14(2):117–135
35. Mäkelä MM, Neittaanmäki P (1992) Nonsmooth optimization: Analysis and algorithms with applications to optimal control. World Scientific, River Edge
36. Mistakidis ES, Stavroulakis GE (1998) Nonconvex optimization in mechanics. Algorithms, heuristics and engineering applications by the FEM. Kluwer, Dordrecht
37. Moreau JJ, Panagiotopoulos PD, Strang G (eds) (1988) Topics in nonsmooth mechanics. Birkhäuser, Basel
38. Outrata J, Kočvara M, Zowe J (1998) Nonsmooth approach to optimization problems with equilibrium constraints. Theory, applications and numerical results. Kluwer, Dordrecht
39. Robinson SM (1999) Linear convergence of epsilon-subgradient descent methods for a class of convex functions. Math Program 86(1):41–50
40. Sagastizábal C, Solodov M (2005) An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. SIAM J Optim 16(1):146–169

41. Schramm H, Zowe J (1992) A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. SIAM J Optim 2(1):121–152
42. Shor NZ (1985) Minimization methods for non-differentiable functions. Springer, Berlin
43. Uryasev SP (1991) New variable-metric algorithms for nondifferentiable optimization problems. J Optim Theory Appl 71(2):359–388
44. Vlček J, Lukšan L (2001) Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization. J Optim Theory Appl 111(2):407–430

Chapter 16
Shape Optimization via Control of a Shape Function on a Fixed Domain: Theory and Numerical Results

Peter Philip and Dan Tiba

Abstract We present a fixed-domain approach for the solution of shape optimization problems governed by linear or nonlinear elliptic partial differential state equations with Dirichlet boundary conditions, where shape optimization is facilitated via optimal control of a shape function. The method involves extending the state equation to a larger domain using regularization. Results regarding the convergence to the original problem are provided as well as differentiability properties of the control-to-state mappings. An algorithm for the numerical implementation of the method is stated and, in a series of numerical shape optimization experiments, the algorithm's behavior is studied with regard to varying the regularization parameter and initial conditions.

16.1 Formulation of the Shape Optimization Problems

We discuss shape optimization problems governed by linear or nonlinear elliptic partial differential equations via a fixed domain approach.

Let $E \subseteq D \subseteq \mathbb{R}^d$, $d \in \mathbb{N}$, be some given bounded domains with a Lipschitzian boundary. Let $\Omega \subseteq D$ be some (unknown) domain and $y \in H_0^1(\Omega)$ be the solution of the following equation defined in Ω:

$$\int_\Omega \left[\sum_{i,j=1}^{d} a_{ij} \frac{\partial y}{\partial x_i} \frac{\partial v}{\partial x_j} + a_0 y v \right] \mathrm{d}x = \int_\Omega f v \mathrm{d}x, \quad \forall v \in H_0^1(\Omega). \tag{16.1}$$

P. Philip (✉)
Department of Mathematics, Ludwig-Maximilians University (LMU) Munich, Theresienstrasse 39, 80333 Munich, Germany
e-mail: philip@math.lmu.de

D. Tiba
Institute of Mathematics, Romanian Academy, PO Box 1-764, 014700 Bucharest, Romania
e-mail: dan.tiba@imar.ro

D. Tiba
Academy of Romanian Scientists, Bucharest, Romania

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_16,

Here $a_{ij}, a_0 \in L^\infty(D)$, $\{a_{ij}\}_{i,j=1,d}$ elliptic and $f \in L^2(D)$.

Alternatively, in Ω, the stationary Navier-Stokes system may be considered:

$$\int_\Omega \left[\eta \sum_{i,j=1}^{d} \frac{\partial y_i}{\partial x_i} \frac{\partial v_j}{\partial x_i} + \sum_{i,j=1}^{d} y_i \frac{\partial y_j}{\partial x_i} v_j \right] \mathrm{d}x = \int_\Omega \sum_{j=1}^{d} f_j v_j \mathrm{d}x,$$
$$\forall v \in V(\Omega),\ y \in V(\Omega). \tag{16.2}$$

Above, $\eta > 0$ is the viscosity, $f_i \in L^2(D)$, and

$$V(\Omega) := \mathrm{cl}_{H_0^1(\Omega)} \big\{ y \in C_0^\infty(\Omega)^d : \mathrm{div}\, y = 0 \big\}, \tag{16.3}$$

where $\mathrm{cl}_{H_0^1(\Omega)}$ denotes the closure in $H_0^1(\Omega)$.

If Ω is Lipschitzian, then $V(\Omega) = \{y \in H_0^1(\Omega)^d : \mathrm{div}\, y = 0\}$ [27], but this is not valid in general since Lions' lemma may fail [7]. See [28] for recent progress in this respect. It is to be noted that the uniqueness is not valid in (16.2).

A general shape optimization problem associated to (16.1) (or to (16.2), (16.3)) consists in the minimization of a cost functional of the form

$$F(y, \Omega) = \int_\Lambda j\big(x, y(x), \nabla y(x)\big) \mathrm{d}x, \tag{16.4}$$

where Λ may be E, Ω, or D and y is the solution of the corresponding state system (extended by 0 to the whole D when $\Lambda = D$). The integrand $j : D \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ (respectively $j : D \times \mathbb{R}^d \times \mathbb{R}^{d\times d} \to \mathbb{R}$) satisfies measurability, continuity, differentiability, and/or convexity assumptions [17, Thm. A3.15].

A special case of (16.4), frequently used in applications, is

$$F(y, \Omega) = \int_\Lambda j\big(x, y(x)\big) \mathrm{d}x, \tag{16.5}$$

with similar notation as above. An important example is the quadratic functional

$$J(\Omega) = \alpha \int_\Lambda |y - y_d|^2 \mathrm{d}x + \beta \int_\Lambda |\nabla y - \nabla y_d|^2 \mathrm{d}x,$$

where $\alpha, \beta \in \mathbb{R}_0^+$, $\alpha + \beta > 0$, $y_d \in H^1(D)$ is given and $|\cdot|$ denotes the modulus or the Euclidean norm in a finite-dimensional space.

Various constraints may be imposed as well. For instance, if $\Lambda = E$, then we impose

$$\Omega \supseteq E \tag{16.6}$$

for any admissible domain Ω and consequently (16.4), (16.5) make sense. State constraints may be added as well. In an abstract form, they are written as $y \in K \subseteq H_0^1(\Omega)$ (respectively $K \subseteq V(\Omega)$), and are usually penalized in the cost functional by adding $(I_K)_\varepsilon(y)$, the Yosida regularization of the indicator function I_K.

As basic references for this paper, we quote [9, 10, 16, 17]. From the point of view of the existence of optimal domains, compactness hypotheses have to be required for the family of admissible domains U_{ad}. In this respect, we mention the uniform class C assumptions, studied in detail in [17, Chap. 2]. A special case is that of uniformly Lipschitzian domains [3], as in (16.2), (16.3). We exemplify with an existence result from [9], valid even if the state equation (16.2), (16.3) has nonunique solution.

Theorem 16.1 *Under the above conditions on $j(\cdot,\cdot,\cdot)$, if U_{ad} is compact with respect to the complementary Hausdorff-Pompeiu metric for open sets, and if there is $\hat{\Omega} \in U_{\text{ad}}$ that together with some solution $\hat{y} \in V(\hat{\Omega})$ of (16.2) satisfies $F(\hat{y}, \hat{\Omega}) < \infty$, then the shape optimization problem (16.2), (16.3), (16.4), (16.6) has at least one optimal pair $(y^*, \Omega^*) \in V(\Omega^*) \times U_{\text{ad}}$.*

In this case, the shape optimization problem (16.2), (16.3), (16.4), (16.6) should be understood in the sense of singular control problems [12].

In the next section, we describe our method and several properties. The second part of the paper is devoted to the algorithm and to a numerical study of its behavior.

16.2 The Fixed Domain Method: Main Results

We introduce the family of admissible domains via a shape function $g \in X(D)$, $X(D)$ being a subspace of piecewise continuous mappings defined in D:

$$\Omega = \Omega_g = \text{int}\{x \in D : g(x) \geq 0\}. \tag{16.7}$$

More precisely, this means that there exists $l \in \mathbb{N}$ and $\Omega_i \subseteq D$, $i \in \{1, \ldots, l\}$, open subsets such that $\Omega_i \cap \Omega_j = \varnothing$, $i \neq j$, $\overline{D} = \bigcup_{i=1}^{l} \overline{\Omega_i}$, and $g_i \in C(\overline{D})$ such that $g|_{\Omega_i} = g_i$ for each $i \in \{1, \ldots, l\}$.

If the constraint (16.6) is imposed, then we require

$$g \geq 0 \quad \text{in } E. \tag{16.8}$$

Functions $g \in X(D)$ are generally called level functions. However, we use the name shape functions since our approach is different from the well-known method of Osher and Sethian [18, 24] (see [2, 13, 29] and references therein for recent advances).

In our approach, the unknown geometry is not considered as "moving" together with a time-like variable and $g \in X(D)$ depends just on the spatial variables. Moreover, no Hamilton-Jacobi equation is needed in our approach. It enters the class of fictitious or embedding methods and it is based on tools from the optimal control theory. One may compare it with the work of Belytschko et al. [1] and Santosa [21] (the second method discussed in that paper), but the first reference in this respect seems to be [14].

An important approximation property, specific to partial differential equations with Dirichlet boundary conditions, is at the base of this method. We denote by $H^\varepsilon : \mathbb{R} \to \mathbb{R}$ a regularization of the Yosida approximation of the maximal monotone extension in $\mathbb{R} \times \mathbb{R}$ of the Heaviside function H. For instance

$$H^\varepsilon(r) := \begin{cases} 1 & \text{for } r \geq 0, \\ \frac{\varepsilon(r+\varepsilon)^2 - 2r(r+\varepsilon)^2}{\varepsilon^3} & \text{for } -\varepsilon < r < 0, \\ 0 & \text{for } r \leq -\varepsilon. \end{cases} \tag{16.9}$$

Other variants are also possible, see [14].

We approximate (16.1) (respectively (16.2)) by

$$\int_D \left[\sum_{i,j=1}^{d} a_{ij} \frac{\partial y_\varepsilon}{\partial x_i} \frac{\partial v}{\partial x_j} + a_0 y_\varepsilon v + \frac{1}{\varepsilon}\big(1 - H^\varepsilon(g)\big) y_\varepsilon v \right] \mathrm{d}x$$

$$= \int_D f v \, \mathrm{d}x, \quad \forall v \in H_0^1(D), \ y_\varepsilon \in H_0^1(D), \tag{16.10}$$

$$\int_D \left[\eta \sum_{i,j=1}^{d} \frac{\partial y_j^\varepsilon}{\partial x_i} \frac{\partial v_j}{\partial x_i} + \sum_{i,j=1}^{d} y_i^\varepsilon \frac{\partial y_j^\varepsilon}{\partial x_i} v_j + \frac{1}{\varepsilon}\big(1 - H^\varepsilon(g)\big) y^\varepsilon \cdot v \right] \mathrm{d}x$$

$$= \int_D f \cdot v \, \mathrm{d}x, \quad \forall v \in V(D), \ y^\varepsilon \in V(D). \tag{16.11}$$

In (16.11), we denote by "$\cdot$" the inner product in $\mathbb{R}^d$. Notice that $H^\varepsilon(g(x)) = 1$ for $g(x) \geq 0$ and $H^\varepsilon(g(x)) = 0$ for $g(x) \leq -\varepsilon$. Therefore, $H^\varepsilon(g)$ defined by (16.9) is an approximation of the characteristic function of Ω_g defined by (16.7). The basic idea of the approximations (16.10), (16.11) is to penalize y_ε, respectively y^ε, outside Ω_g. A similar idea was used for the first time by Kawarada and Natori [15].

In the remainder of this section, we fix $d = 3$.

Theorem 16.2

(i) *If $\Omega = \Omega_g$ is of class C, then $y_\varepsilon|_{\Omega_g} \to y_g$ weakly in $H^1(\Omega_g)$.*
(ii) *If $\eta > 0$ is big with respect to $|f|_{V(D)^*}$, then the solution of (16.2) is unique. If $\Omega = \Omega_g$ is Lipschitzian, then $y^\varepsilon|_{\Omega_g} \to y^g$ weakly in $H^1(\Omega_g)^3$.*

For proofs and more details, we quote [9, 16]. The hypothesis $\eta \geq c|f|_{V(D)^*}$ with $c > 0$ "big", ensures the uniqueness of the solution in (16.2), respectively (16.11).

This approximation allows studying the optimization problems (16.10), (16.4), (16.6), respectively (16.11), (16.4), (16.6). The following differentiability properties play an outstanding role:

Theorem 16.3

(i) *The mapping* $g \mapsto y_\varepsilon = y_\varepsilon(g)$ *defined by* (16.10) *is Gâteaux differentiable between* $X(D)$ *and* $H_0^1(D)$ *and* $z = \nabla y_\varepsilon(g)w$ *satisfies the equation in variations*:

$$\int_D \left[\sum_{i,j=1}^{3} a_{ij} \frac{\partial z}{\partial x_i} \frac{\partial v}{\partial x_j} + a_0 z v + \frac{1}{\varepsilon}\big(1 - H^\varepsilon(g)\big) z v \right] \mathrm{d}x$$
$$= \frac{1}{\varepsilon} \int_D \big(H^\varepsilon\big)'(g) w y_\varepsilon v \mathrm{d}x, \quad \forall v \in H_0^1(D),\ z \in H_0^1(D). \tag{16.12}$$

(ii) *The mapping* $g \mapsto y^\varepsilon = y^\varepsilon(g)$ *defined by* (16.11) *is Gâteaux differentiable between* $X(D)$ *and* $V(D)$ *and the derivative in the direction* $w \in X(D)$, *denoted by* $z = (z_1, z_2, z_3) \in V(D)$, *satisfies the equation in variations*:

$$\int_D \left[\eta \sum_{i,j=1}^{3} \frac{\partial z_j}{\partial x_i} \frac{\partial v_j}{\partial x_j} + \sum_{i,j=1}^{3} y_i^\varepsilon \frac{\partial z_j}{\partial x_i} v_j + \sum_{i,j=1}^{3} z_i \frac{\partial y\varepsilon_j}{\partial x_i} v_j \right] \mathrm{d}x$$
$$+ \frac{1}{\varepsilon} \int_D \big(1 - H^\varepsilon(g)\big) z \cdot v \mathrm{d}x$$
$$= \frac{1}{\varepsilon} \int_D \big[\big(H^\varepsilon\big)'(g) w\big] y^\varepsilon \cdot v \mathrm{d}x, \quad \forall v \in V(D),\ z \in V(D). \tag{16.13}$$

Starting from (16.12), (16.13) we can introduce the corresponding adjoint equations that are used in the computation of the gradient of the cost functional with respect to $g \in X(D)$, [9, 16]. Examples of this type are detailed in the second part of this article. We underline that similar approximation procedures are very useful in free boundary problems as well [8].

Remark 16.1 The variations used to prove Theorem 16.3 are of the form $g + \lambda w$, $\lambda \in \mathbb{R}$, $g, w \in X(D)$. They are called functional variations in [16] and allow for simultaneous changes of the boundary and of the topological characteristic of the searched domain.

16.3 Numerical Experiments

16.3.1 Setting and Numerical Methods

For all the numerical experiments presented below, we used the square fixed domain $D :=]-1, 1[\times]-1, 1[\subseteq \mathbb{R}^2$ with the fixed subdomain $E :=]-\frac{1}{2}, \frac{1}{2}[\times]-\frac{1}{2}, \frac{1}{2}[\subseteq D$, where the shape function constraint (16.8), restated as

$$g \in U(D) := \big\{ g \in X(D) : g \geq 0 \text{ on } E \big\}, \tag{16.14}$$

is used in the numerical examples below, where indicated.

In each experiment, the state equation for $y_\varepsilon \in H_0^1(D)$ is a special case of (16.10), having the form

$$\int_D \left[\frac{\partial y_\varepsilon}{\partial x_1} \frac{\partial v}{\partial x_1} + \frac{\partial y_\varepsilon}{\partial x_2} \frac{\partial v}{\partial x_2} + \frac{1}{\varepsilon}\big(1 - H^\varepsilon(g)\big) y_\varepsilon v \right] \mathrm{d}x = \int_D f v \, \mathrm{d}x, \quad \forall v \in H_0^1(D), \tag{16.15}$$

with a fixed right-hand side $f \equiv 1$ (except in Example 16.3), and where the influence of the regularization parameter $\varepsilon > 0$ is among the aspects subsequently investigated.

The cost functionals considered for the shape optimization have the general form

$$J : X(D) \to \mathbb{R}, \qquad g \mapsto J(g) = F\big(S(g), g\big), \tag{16.16}$$

where $F : H_0^1(D) \times X(D) \to \mathbb{R}$ and $S : X(D) \to H_0^1(D)$, $S(g) = y_\varepsilon(g)$, is the control-to-state operator corresponding to (16.15).

The shape optimization algorithm used here is structurally the same that was previously described in [16, Sect. 5]. The algorithm is restated for the convenience of the reader, which also provides us with the opportunity to elaborate on the individual steps of the algorithm, where appropriate.

Algorithm 16.1 Shape optimization

Step 1 Set $n := 0$ and choose an admissible initial shape function $g_0 \in X(D)$.

Step 2 Compute the solution to the state equation $y_n = y_\varepsilon = S(g_n)$ (note that $\varepsilon > 0$ is fixed throughout the algorithm).

Step 3 Compute the solution to the corresponding adjoint equation $p_n = p_\varepsilon$.

Step 4 Compute a descent direction $w_{\mathrm{d},n} = w_{\mathrm{d},n}(y_n, p_n)$.

Step 5 Set $\tilde{g}_n := g_n + \lambda_n w_{\mathrm{d},n}$, where $\lambda_n \geq 0$ is determined via line search, i.e. as a solution to the minimization problem

$$\lambda \mapsto J(g_n + \lambda w_{\mathrm{d},n}) \to \min. \tag{16.17}$$

We have implemented a golden section search [20, Sect. 10.2] to numerically carry out the minimization (16.17). Note that the minimization (16.17) is typically nonconvex and the golden section search will, in general, only provide a *local* min λ_n. As usual, this can be alleviated by stochastically varying the initial guess for the line search (no such stochastic variation has been used for the computational results presented below).

Step 6 Set $g_{n+1} := \pi_{U(D)}(\tilde{g}_n)$, where $\pi_{U(D)}$ denotes the projection

$$\pi_{U(D)} : X(D) \to U(D), \quad \pi_{U(D)}(g)(x) := \begin{cases} \max\{0, g(x)\} & \text{for } x \in E, \\ g(x) & \text{for } x \in D \setminus E \end{cases} \tag{16.18}$$

(and $U(D) = X(D)$, $\pi_{U(D)}(g) = g$ if no constraints are imposed).

Step 7 RETURN $g_{\text{fin}} := g_{n+1}$ if the change of g and/or the change of $J(g)$ are below some prescribed tolerance parameter. Otherwise: Increment n, i.e. $n := n + 1$ and GO TO Step 2. Clearly, one can think of several reasonable quantities for measuring the change of g and/or the change of $J(g)$. For all the numerical examples discussed subsequently, we stopped the iteration and returned $g_{\text{fin}} := g_{n+1}$ if $\|g_n - g_{n+1}\|_2 < 10^{-8}$ OR $|J(g_n) - J(g_{n+1})| < 10^{-8}$, where $\|g_n - g_{n+1}\|_2 / \|g_{n+1}\|_2$ is used instead of $\|g_n - g_{n+1}\|_2$ if $\|g_{n+1}\|_2 > 1$ and $|J(g_n) - J(g_{n+1})| / |J(g_{n+1})|$ is used instead of $|J(g_n) - J(g_{n+1})|$ if $|J(g_{n+1})| > 1$.

The state equations as well as the adjoint equations that need to be solved numerically during Algorithm 16.1 are linear elliptic PDEs with homogeneous Dirichlet boundary conditions. The numerical solution is obtained via a finite volume discretization [19, Sect. 4], [4, Chap. III]. More precisely, the software *WIAS-HiTNIHS*,[1] originally designed for the solution of more general PDEs occurring when modeling conductive-radiative heat transfer and electromagnetic heating [6, 11], has been adapted for use in the present context. *WIAS-HiTNIHS* is based on the program package *pdelib* [5], it employs the grid generator *Triangle* [25, 26] to produce constrained Delaunay triangulations of the domains, and it uses the sparse matrix solver *PARDISO* [22, 23] to solve the linear system arising from the finite volume scheme.

Except where indicated otherwise, we use a fixed triangular grid provided by *Triangle*, consisting of 31168 triangles.

The numerical scheme yields discrete approximations of y_n and p_n (cf. Steps 2 and 3 of Algorithm 16.1), defined at each vertex of the triangular discrete grid, interpolated piecewise affinely, i.e. affinely to each triangle of the discrete grid. In consequence, the (approximate) shape functions g_n are piecewise affine as well. Where integrals of these piecewise affine functions need to be computed (e.g., in Step 7 of Algorithm 16.1), they are computed exactly.

16.3.2 Examples of Numerical Shape Optimizations

Example 16.1 As explained above, in the present paper, we use a numerical solver different from the one used in [16]. Thus, for verification purposes, we begin by applying the solver to the situation of [16, Example 1], i.e. the cost functional J is as in (16.16) with

$$F(y, g) := \frac{1}{2} \int_E (y - y_d)^2 \, \mathrm{d}x, \tag{16.19a}$$

$$y_d(x_1, x_2) := -\left(x_1 - \frac{1}{2}\right)^2 - \left(x_2 - \frac{1}{2}\right)^2 + \frac{1}{16}, \tag{16.19b}$$

[1] **Hi**gh **T**emperature **N**umerical **I**nduction **H**eating **S**imulator; pronunciation: ∼hit-nice.

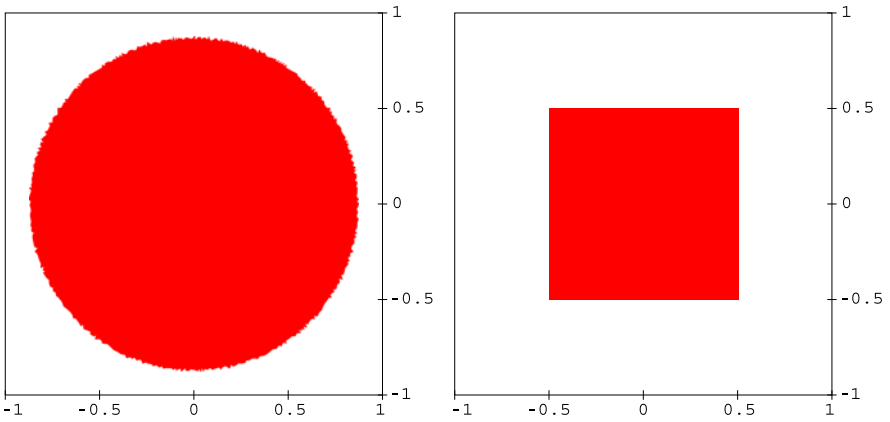


Fig. 16.1 Shapes Ω_{g_0} (on the *left*) and $\Omega_{g_{\text{fin}}}$ (on the *right*) for the shape optimization of Example 16.1 with g_0 according to (16.21)

the adjoint equation for the adjoint state $p_\varepsilon \in H_0^1(D)$ is

$$\int_D \left[\frac{\partial p_\varepsilon}{\partial x_1}\frac{\partial v}{\partial x_1} + \frac{\partial p_\varepsilon}{\partial x_2}\frac{\partial v}{\partial x_2} + \frac{1}{\varepsilon}\big(1 - H^\varepsilon(g)\big)p_\varepsilon v \right] \mathrm{d}x$$
$$= \int_E (y_\varepsilon - y_d)v \,\mathrm{d}x, \quad \forall v \in H_0^1(D), \tag{16.20}$$

the descent direction used in Step 4 is $w_{\mathrm{d}}(y, p) = -\frac{1}{\varepsilon}yp$, $\varepsilon = 10^{-5}$, $g \geq 0$ on E is imposed, and the initial shape function is

$$g_0(x_1, x_2) := \frac{3}{4} - x_1^2 - x_2^2. \tag{16.21}$$

The resulting initial cost is $J(g_0) = 0.370026$ and the final cost $J(g_{\text{fin}}) = 0.294218$. Using a refined grid with 124593 triangles yields $J(g_0) = 0.369105$ and $J(g_{\text{fin}}) = 0.292884$. These values are, respectively, some 2 % and 9 % larger than the corresponding values in [16, Example 1], most likely due to different methods used for numerically computing the integrals for the cost functional. For the initial guess for the line search of Step 5, we use $\lambda = 1$. As it turns out, during the very first line search, this already yields $\tilde{g}_1 < 0$ on all of D, and the corresponding cost is $J(\tilde{g}_1) = 0.271456$. The projection g_1 then already corresponds to the square as depicted in Fig. 16.1, and the cost is almost identical to the final cost. After the second line search, the relative change in the L^2-norm of g is 2.1×10^{-6} and the absolute change in the corresponding costs is 1.7×10^{-16}, i.e. the iteration is stopped according to the criterion in Step 7.

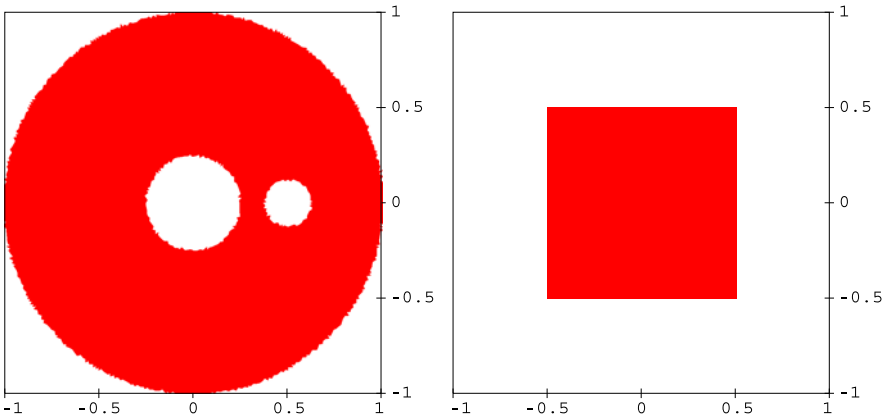


Fig. 16.2 Shapes Ω_{g_0} (on the *left*) and $\Omega_{g_{\text{fin}}}$ (on the *right*) for the shape optimization of Example 16.1 with g_0 according to (16.22)

Next, we assess the stability of Algorithm 16.1 with regard to different initial shape functions g_0. If we use the function g_0 from [16, Example 2], i.e.

$$g_0(x_1, x_2) := \min\left\{x_1^2 + x_2^2 - \frac{1}{16},\ \left(x_1 - \frac{1}{2}\right)^2 + x_2^2 - \frac{1}{64},\ 1 - x_1^2 - x_2^2\right\}, \quad (16.22)$$

then $J(g_0) = 0.301027$ and $J(g_{\text{fin}}) = 0.294218$, and Algorithm 16.1 performs precisely as before (see Fig. 16.2). If we use

$$g_0(x_1, x_2) := 1, \quad (16.23)$$

then, $J(g_0) = 0.454657$ and $J(g_{\text{fin}}) = 0.294218$, and the only difference now lies in g remaining positive throughout in a small neighborhood of the outer boundary. Even though the convergence criterion of Step 7 is, again, satisfied after the second line search, g_{fin} still has this property, as can be seen near the corners in the right-hand picture of Fig. 16.3. Actually, it is clear that g can never change its sign on the outer boundary ∂D, since the descent direction is $w_{\text{d}}(y_n, p_n) = -\frac{1}{\varepsilon} y_n p_n$ and y_n and p_n are fixed to 0 on ∂D via the Dirichlet boundary condition.

We now, once again, use g_0 from (16.21), but now we vary the regularization parameter ε. We observe a stable performance over many orders of magnitude. Values for the initial and final costs depending on ε are compiled in Table 16.1, where, in every case, the picture is as in Fig. 16.1, and convergence is always obtained after the second line search. Of course, Theorem 16.2 provides $y^\varepsilon|_{\Omega_g} \to y^g$ for $\varepsilon \to 0$. The expected convergence of $J(g_{\text{fin}})$ for $\varepsilon \to 0$ can, indeed, be observed in the values provided in Table 16.1.

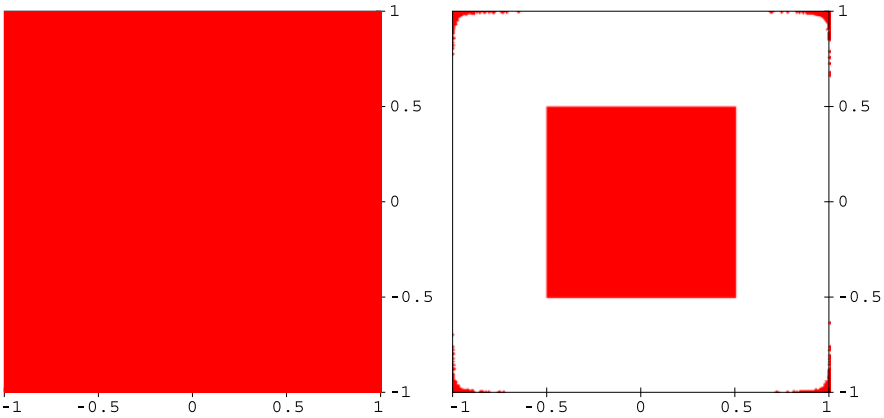


Fig. 16.3 Shapes Ω_{g_0} (on the *left*) and $\Omega_{g_{\text{fin}}}$ (on the *right*) for the shape optimization of Example 16.1 with g_0 according to (16.23)

Table 16.1 Dependence of the initial and final cost on ε in Example 16.1 with g_0 from (16.21)

ε	$J(g_0)$	$J(g_{\text{fin}})$
10^{-13}	0.369777	0.294085
10^{-9}	0.369777	0.294085
10^{-5}	0.370026	0.294218
10^{-2}	0.404680	0.316894

Example 16.2 For this example, we fix $\varepsilon = 10^{-5}$. We now use a different cost functional. As before, J has the form (16.16), but now with

$$F(y, g) := \int_D H^\varepsilon(g)(y - y_d)\,\mathrm{d}x, \tag{16.24a}$$

$$y_d(x_1, x_2) := -\left(x_1 - \frac{1}{2}\right)^2 - \left(x_2 - \frac{1}{2}\right)^2 + \frac{1}{8}, \tag{16.24b}$$

and the adjoint equation is

$$\int_D \left[\frac{\partial p_\varepsilon}{\partial x_1}\frac{\partial v}{\partial x_1} + \frac{\partial p_\varepsilon}{\partial x_2}\frac{\partial v}{\partial x_2} + \frac{1}{\varepsilon}\big(1 - H^\varepsilon(g)\big)p_\varepsilon v\right]\mathrm{d}x$$
$$= \int_D H^\varepsilon(g)v\,\mathrm{d}x, \quad \forall v \in H_0^1(D). \tag{16.25}$$

Note that (16.24a) is an approximation for $\int_\Omega (y - y_d)\mathrm{d}x$, cf. [16, Eqs. (26), (27)]. Also note that, in contrast to the F from (16.19a), (16.19b), the current F can become negative (as, indeed, it does in the computational examples below). For the

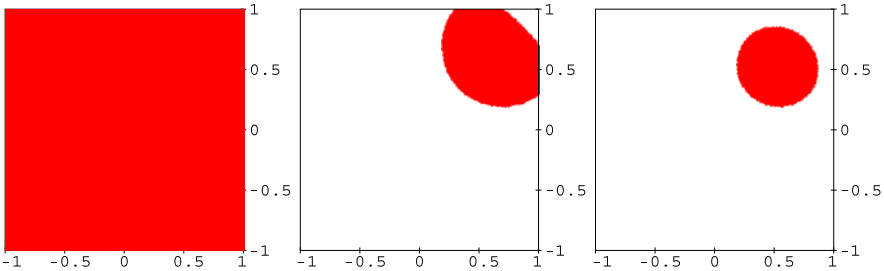


Fig. 16.4 Shape optimization of Example 16.2 with g_0 according to (16.23): Ω_{g_0} (on the *left*, $J(g_0) = 4.72921$), an intermediate shape during line search #1 (*middle*, cost 0.00287023), $\Omega_{g_{\text{fin}}}$ (on the *right*, $J(g_{\text{fin}}) = -0.0187939$)

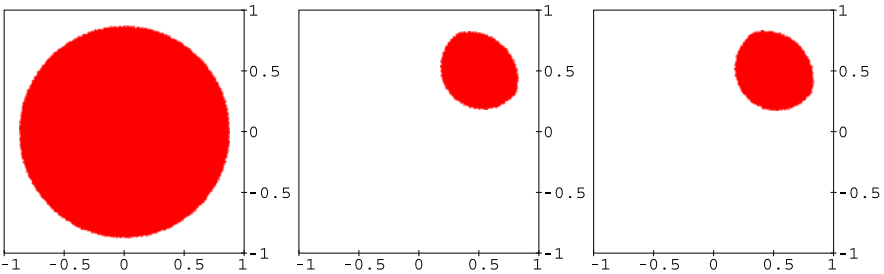


Fig. 16.5 Shape optimization of Example 16.2 with g_0 according to (16.21): Ω_{g_0} (on the *left*, $J(g_0) = 1.99556$), Ω_{g_1} (*middle*, $J(g_1) = -0.0191348$), $\Omega_{g_{\text{fin}}}$ (on the *right*, $J(g_{\text{fin}}) = -0.0191523$)

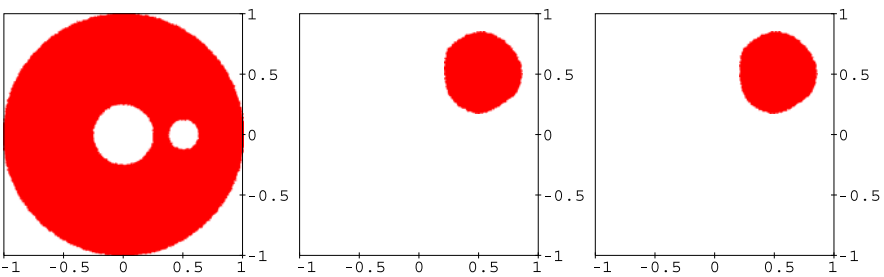


Fig. 16.6 Shape optimization of Example 16.2 with g_0 according to (16.22): Ω_{g_0} (on the *left*, $J(g_0) = 2.78662$), Ω_{g_1} (*middle*, $J(g_1) = -0.0189457$), $\Omega_{g_{\text{fin}}}$ (on the *right*, $J(g_{\text{fin}}) = -0.0189734$)

descent direction of Step 4, we use (cf. [16, Remark 6]):

$$w_{\text{d}}(y, p) = -\left(H^{\varepsilon}(g)(y - y_d) + \frac{1}{\varepsilon} yp\right). \tag{16.26}$$

In the last experiment of Example 16.1, depicted in Fig. 16.3, we found a noticeable dependence of $\Omega_{g_{\text{fin}}}$ on the initial shape function g_0. In the present situation, we once again find that the final shape $\Omega_{g_{\text{fin}}}$ depends slightly but noticeably on g_0: We ran computations with g_0 according to (16.23), (16.21), and (16.22), respectively, and the results are illustrated in Figs. 16.4–16.6. In each case, convergence according to the Step 7 criterion occurred after three line searches. Even though similar, the final shapes differ slightly in the three situations, as can be seen by comparing the pictures on the right in Figs. 16.4–16.6. The corresponding values of the cost functional are provided in the respective figure captions.

In Example 16.1, we noted that the sign of g was always fixed on ∂D. However, in the present situation, the sign of g on ∂D can and does change due to the additional term in (16.26).

We also note that (16.24a), (16.24b), (16.25), and (16.26) are all symmetric with respect to exchanging x_1 and x_2. As expected, this symmetry can be observed in the shapes in Figs. 16.4 and 16.5, where the initial condition satisfies the same symmetry. The symmetry is slightly broken in Fig. 16.6 due to the initial condition.

Example 16.3 For this example, we once again fix $\varepsilon = 10^{-5}$. In contrast to all the previous examples, we use a nonconstant right-hand side, namely

$$f : D \to \mathbb{R}, \qquad f(x_1, x_2) := -x_1^2 x_2^2 + 1. \tag{16.27}$$

The cost functional is similar to, but different from, the one used in Example 16.1: Now J has the form (16.16) with

$$F(y, g) := \frac{1}{2} \int_D (y - y_d)^2 \, \mathrm{d}x, \tag{16.28a}$$

$$y_d(x_1, x_2) := x_1^2 x_2^2. \tag{16.28b}$$

Note that y_d is different from the y_d in (16.19b) and, in contrast to (16.19a), in integration in (16.28a) is over all of D. The adjoint equation is

$$\begin{aligned} &\int_D \left[\frac{\partial p_\varepsilon}{\partial x_1} \frac{\partial v}{\partial x_1} + \frac{\partial p_\varepsilon}{\partial x_2} \frac{\partial v}{\partial x_2} + \frac{1}{\varepsilon} \big(1 - H^\varepsilon(g)\big) p_\varepsilon v \right] \mathrm{d}x \\ &\quad = \int_D (y_\varepsilon - y_d) v \, \mathrm{d}x, \quad \forall v \in H_0^1(D). \end{aligned} \tag{16.29}$$

As in Example 16.1, for Step 4, we use the descent direction $w_{\mathrm{d}}(y, p) = -\frac{1}{\varepsilon} yp$.

In the present situation, the nonconvexity of the problem is much more visible in the numerical results than during previous examples. We observe a considerable dependence of the final shape not only on the initial shape function g_0 but also on the initial guess for λ during the line searches. As for the previous examples, we used the convergence criterion of Step 7. In Figs. 16.7 and 16.8, we show the results for shape optimizations using g_0 according to (16.23) and (16.22), respectively, using $\lambda = 1$ for the initial guess. We have also independently tested $\lambda = 1000$ for g_0 according

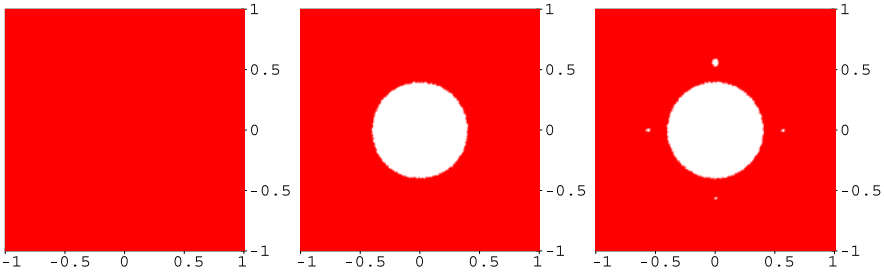


Fig. 16.7 Shape optimization of Example 16.3 with g_0 according to (16.23) and an initial guess for the line search $\lambda = 1$: Ω_{g_0} (on the *left*, $J(g_0) = 0.0795982$), the shape after line search #1 (*middle*, $J(g_1) = 0.0721385$), $\Omega_{g_{\text{fin}}} = \Omega_{g_5}$ (on the *right*, $J(g_{\text{fin}}) = J(g_5) = 0.0720632$, $\|g_{\text{fin}}\|_2 = 5.71677$)

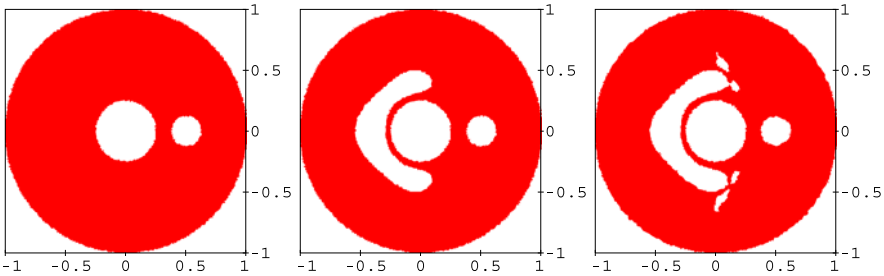


Fig. 16.8 Shape optimization of Example 16.3 with g_0 according to (16.22) and an initial guess for the line search $\lambda = 1$: Ω_{g_0} (on the *left*, $J(g_0) = 0.0791799$), the shape after line search #1 (*middle*, $J(g_1) = 0.0781853$), $\Omega_{g_{\text{fin}}} = \Omega_{g_{13}}$ (on the *right*, $J(g_{\text{fin}}) = J(g_{13}) = 0.0781504$, $\|g_{\text{fin}}\|_2 = 4.04025$)

to (16.22) and we have found another local minimum, see Fig. 16.9. As noted in the respective figure captions, the number of line searches needed varied between 2 and 17, and the L^2-norm of the final shape function varied between 4 (sic) and 10^8.

As in Example 16.1, the sign of g on ∂D cannot change during the algorithm, and it actually does not. Figure 16.9 is deceiving in this regard, due to the fact that the color of each triangle of the discretization is determined by the *average* of g on the respective triangle.

In the present situation, we have x_1-x_2-symmetry as well as symmetry with respect to the signs of x_1 and x_2, respectively, provided that the initial shape function satisfies the same symmetry. These symmetries are visible in Fig. 16.7, slightly broken in the final shape due to the discrete grid.

Remark 16.2 In Examples 16.1 and 16.3, it seems advisable to use our descent direction $w_{\text{d}}(y, p) = -\frac{1}{\varepsilon} yp$, rather than the $w_{\text{d}}(y, p) = -\frac{1}{\varepsilon}(H^\varepsilon)'(g)yp$ mentioned in [16, (26)]—for example, for $g_0 = 1$, $(H^\varepsilon)'(g_0) = 0$ everywhere on D, and one is stuck at the initial shape function. Numerically, the same remained true for all other g_0 we tested. To make use of the w_{d} from [16, (26)], one would need to

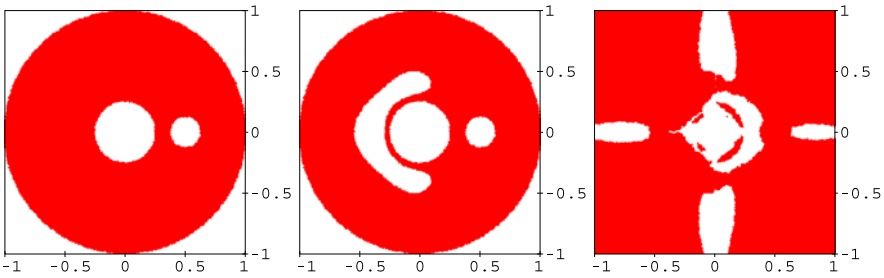


Fig. 16.9 Shape optimization of Example 16.3 with g_0 according to (16.22) and an initial guess for the line search $\lambda = 1000$: Ω_{g_0} (on the *left*, $J(g_0) = 0.0791799$), the shape after line search #1 (*middle*, $J(g_1) = 0.0781853$), $\Omega_{g_{\text{fin}}} = \Omega_{g_{17}}$ (on the *right*, $J(g_{\text{fin}}) = J(g_{17}) = 0.0726054$, $\|g_{\text{fin}}\|_2 = 1.04 \times 10^8$)

tremendously refine the grid in the typically small set $\{x \in D : -\varepsilon < g(x) < 0\}$, where $(H^\varepsilon)'(g)$ is nonzero. Respectively, the same remark applies with respect to the descent direction used in Example 16.2, cf. [16, Remark 6].

16.4 Conclusions

The fixed-domain method described in the paper allows one to make use of optimal control techniques in the context of shape optimization problems governed by elliptic partial differential state equations. The control is given in the form of a shape function encoding the unknown optimal domain, where the shape function is introduced into the state equations by composing it with a regularized Heaviside function and penalization, such that the state is penalized outside the domain encoded by the shape function. The approach is supported by rigorous results showing the convergence of the regularized problem, given the regularization parameter tends to 0. The established Gâteaux differentiability of the control-to-state map and the availability of the adjoint equation provide the foundation for the presented numerical algorithm. In a series of numerical experiments, the algorithm proves to be effective in solving shape optimization problems. We underline that the observed convergence was rapid, with just one required line search in several examples, and at most 17 required line searches in the most complicated example. The setup of the formulation is such that topology changes can occur naturally during the optimization process. In some of the numerical computations, we find that small differences in the values of the objective functional can correspond to significant shape and topology changes. The experiments suggest that nonconvexity and local minima can be an issue, which we intend to address by adding a stochastic component to the algorithm in future work.

Acknowledgements The work of Dan Tiba was supported by CNCS Romania under Grant ID-PCE-2011-3-0211.

References

1. Belytschko T, Xiao SP, Parimi C (2003) Topology optimization with implicit functions and regularization. Int J Numer Methods Eng 57(8):1177–1196
2. Chen J, Shapiro V, Suresh K, Tsukanov I (2007) Shape optimization with topological changes and parametric control. Int J Numer Methods Eng 71(3):313–346
3. Chenais D (1975) On the existence of a solution in a domain identification problem. J Math Anal Appl 52(2):189–219
4. Eymard R, Gallouët T, Herbin R (2000) Finite volume methods. In: Ciarlet PG, Lions J-L (eds) Handbook of numerical analysis, Vol. VII. North-Holland, Amsterdam, pp 713–1020
5. Fuhrmann J, Koprucki Th, Langmach H (2001) pdelib: An open modular tool box for the numerical solution of partial differential equations. Design patterns. In: Hackbusch W, Wittum G (eds) Proceedings of the 14th GAMM seminar on concepts of numerical software, Kiel, 1998. Vieweg, Braunschweig
6. Geiser J, Klein O, Philip P (2007) Numerical simulation of temperature fields during the sublimation growth of SiC single crystals, using WIAS-HiTNIHS. J Cryst Growth 303(1):352–356
7. Geymonat G, Gilardi G (1998) Contre-exemples à l'inégalité de Korn et au lemme de Lions dans des domaines irréguliers. In: Équations aux dérivées partielles et applications: articles dédiées à J-L Lions. Gauthier-Villars, Paris, pp 541–548
8. Halanay A, Murea C, Tiba D (2012) Existence and approximation for a steady fluid-structure interaction problem using fictitious domain approach with penalization. Submitted to J. Math. Fluid Mech
9. Halanay A, Tiba D (2009) Shape optimization for stationary Navier-Stokes equations. Control Cybern 38(4B):1359–1374
10. Henrot A, Pierre M (2005) Variation et optimisation de formes. Springer, Berlin
11. Klein O, Lechner Ch, Druet P-É, Philip P, Sprekels J, Frank-Rotsch Ch, Kießling F-M, Miller W, Rehse U, Rudolph P (2009) Numerical simulations of the influence of a traveling magnetic field, generated by an internal heater-magnet module, on liquid encapsulated Czochralski crystal growth. Magnetohydrodynamics 45(4):557–567
12. Lions J-L (1983) Contrôle des systèmes distribués singuliers. Méthodes Mathématiques de l'Informatique, vol 13. Gauthier-Villars, Montrouge
13. Luo Z, Wang MY, Wang S, Wei P (2008) A level set-based parametrization method for structural shape and topology optimization. Int J Numer Methods Eng 76(1):1–26
14. Mäkinen RAE, Neittaanmäki P, Tiba D (1992) On a fixed domain approach for a shape optimization problem. In: Amesm W, van Houwen P (eds) Computational and applied mathematics II, Dublin, 1991. North-Holland, Amsterdam, pp 317–326
15. Natori M, Kawarada H (1981) An application of the integrated penalty method to free boundary problems of Laplace equation. Numer Funct Anal Optim 3(1):1–17
16. Neittaanmäki P, Pennanen A, Tiba D (2009) Fixed domain approaches in shape optimization problems with Dirichlet boundary conditions. Inverse Probl 25(5):055003
17. Neittaanmäki P, Sprekels J, Tiba D (2006) Optimization of elliptic systems. Theory and applications. Springer, New York
18. Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. J Comput Phys 79(1):12–49
19. Philip P (2010) Analysis, optimal control, and simulation of conductive-radiative heat transfer. Ann Acad Rom Sci Ser Math Appl 2(2):171–204
20. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007) Numerical recipes: the art of scientific computing, 3rd edn. Cambridge University Press, Cambridge
21. Santosa F (1995/96) A level-set approach for inverse problems involving obstacles. ESAIM Contrôle Optim Calc Var 1:17–33
22. Schenk O, Gärtner K (2004) Solving unsymmetric sparse systems of linear equations with PARDISO. Future Gener Comput Syst 20(3):475–487
23. Schenk O, Gärtner K, Fichtner W (2000) Efficient sparse *LU* factorization with left-right looking strategy on shared memory multiprocessors. BIT Numer Math 40(1):158–176

24. Sethian JA (1996) Level set methods. Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science. Cambridge University Press, Cambridge
25. Shewchuk JR (1996) Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In: Lin MC, Manocha D (eds) Applied computational geometry: towards geometric engineering. Lecture notes in computer science, vol 1148. Springer, Berlin, pp 203–222
26. Shewchuk JR (2002) Delaunay refinement algorithms for triangular mesh generation. Comput Geom 22(1–3):21–74
27. Temam R (1979) Navier-Stokes equations. Theory and numerical analysis. North-Holland, Amsterdam
28. Wang G, Yang D (2008) Decomposition of vector-valued divergence free Sobolev functions and shape optimization for stationary Navier-Stokes equations. Commun Partial Differ Equ 33(1–3):429–449
29. Wang S, Wang MY (2006) Radial basis functions and level set method for structural topology optimization. Int J Numer Methods Eng 65(12):2060–2090

Chapter 17
Multi-Objective Actuator Placement Optimization for Local Sound Control Evaluated in a Stochastic Domain

Tuomas Airaksinen and Timo Aittokoski

Abstract A method to find optimal locations and properties of anti-noise actuators in a local noise control system is considered. The local noise control performance is approximated by an approach based on a finite element method, attempting to estimate the average performance of an optimal active noise control (ANC) system. Local noise control uses a fixed number of circular actuators that are located on the boundary of a three-dimensional enclosed acoustic space. Actuator signals are used to minimize the known harmonic noise at specified locations. The average noise reduction is maximized at two frequency ranges by adjusting the anti-noise actuator configuration, which is a non-linear multi-objective optimization problem. To solve the optimization problem, an unsorted population size evolutionary optimization algorithm (UPS-EMOA) is considered, and its performance is compared to the widely-known NSGA-II method. As a numerical example problem, the ANC in a passenger car cabin is considered. Significantly better noise control is obtained with the optimized actuator locations than only by a engineer's sophisticated guess.

17.1 Introduction

Noise generated by different machines is an increasing problem in modern working environments. Wheels, engines, and cooler fans are typical noise sources. There is an obvious need for noise control applications in factory environments, engineering vehicles, and passenger cars, for example. Sometimes it is possible to remove or reduce important noise source mechanisms by suitable design choices, which makes particular noise control approaches unnecessary. In many cases, however, this is not possible or the design is limited by other more important factors than noise.

Passive noise control techniques such as absorbing and insulating acoustic elements are effective methods in reducing high frequency sound, whereas active noise

T. Airaksinen (✉) · T. Aittokoski
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland
e-mail: tuomas.airaksinen@jyu.fi

T. Aittokoski
e-mail: timo.aittokoski@jyu.fi

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_17,

control (ANC) methods [11] are good at reducing low-frequency noise. The ANC is based on generating anti-sound with actuators. So the original noise is attenuated. In order to cancel the noise perfectly, the anti-sound must have the same amplitude as the noise, but an opposite phase so that destructive interference occurs. Local noise control employs ANC methods so that noise is reduced locally in a desired subdomain.

The most important frequencies originating in the passenger car engine are below 500 Hz [14]. As there are significant low-frequency noise sources, the local sound control can provide a significant noise reduction to the car cabin environment. Advanced methods designing and assessing such systems employ numerical simulation and optimization. Approaches using finite element modeling are presented in the articles [1, 5, 13, 15] of which [5, 13] consider also optimizing locations for anti-noise actuators.

In [1], a numerical evaluation method is developed for optimal local noise control, based on finite element modeling. The method determines the optimal performance of a local sound control by including the stochasticity of the cavity domain in the model. The anti-noise is optimized by minimizing the expected value of the noise computed using the finite element method. In this paper, this method is used to develop a technique to find optimal locations for anti-noise actuators. The optimization of actuator configuration is formulated as a multi-objective optimization problem such that optimal noise reduction at appropriate frequency ranges forms objective functions. By solving a multi-objective optimization problem, a whole family of Pareto-optimal solutions is obtained. An unrestricted population-size evolutionary multi-objective algorithm (UPS-EMOA, [3]) is used to solve the multi-objective optimization problem, and its performance is also compared to a well-known elitist non-dominated sorting genetic algorithm (NSGA-II, [8]).

This article is organized as follows. In Sect. 17.2, a mathematical model of sound propagation, the Helmholtz partial differential equation, and a numerical method to solve it are briefly presented. In Sect. 17.3, the local noise control in a stochastic domain is formulated as a quadratic optimization problem and an example of local noise control in a car driver's ears is described. The objective functions are also derived to evaluate actual anti-noise configurations. In Sect. 17.4, the multi-objective optimization methods used in actuator configuration optimization are described briefly and the used parameters are given. In Sect. 17.5, the numerical results of actuator configuration optimization in a three-dimensional car cabin problem are studied and analyzed. Finally, in Sect. 17.6, conclusions are given.

17.2 An Acoustic Model

The time harmonic sound propagation is modeled by the Helmholtz equation

$$-\nabla \cdot \frac{1}{\rho} \nabla p - \frac{\omega^2}{c^2 \rho} p = 0 \quad \text{in } \Omega, \tag{17.1}$$

where $\rho(\mathbf{x})$ is the density of the material at the location $\mathbf{x}$, and $c(\mathbf{x})$ is the speed of sound in the material. The complex pressure $p(\mathbf{x})$ defines the amplitude and phase of the pressure. The sound pressure at time t is obtained by $\mathfrak{Re}(e^{-i\omega t}p)$, where ω is the angular frequency of sound and $i = \sqrt{-1}$. A sound source f acting on a part S of the boundary $\partial\Omega$ is modeled via a boundary condition. A partially absorbing wall material is described by the impedance boundary conditions

$$\begin{aligned} \frac{\partial p}{\partial \mathbf{n}} &= \frac{i\eta\omega}{c}p + f \quad \text{on } S, \\ \frac{\partial p}{\partial \mathbf{n}} &= \frac{i\eta\omega}{c}p \qquad \text{on } \partial\Omega \setminus S, \end{aligned} \tag{17.2}$$

where $\eta(\mathbf{x})$ is the absorption coefficient depending on the properties of the surface material. The value $\eta = 1$ approximates a perfectly absorbing material and the value $\eta = 0$ approximates a sound-hard material (the Neumann boundary condition).

An approximate solution for the partial differential equation (PDE) (17.1) can be obtained using a finite element method [16]. The finite element discretization transforms (17.1) into a system of linear equations $\mathbf{Ax} = \mathbf{b}$, where the matrix $\mathbf{A}$ is generally symmetric, large, and sparse. Due to the large size and structure of $\mathbf{A}$, direct solution methods are computationally too expensive. Instead, an iterative solution method like GMRES needs to be used. Solving the system with a reasonable number of iterations is, however, challenging as the matrix $\mathbf{A}$ is badly conditioned and especially so when the calculation domain is large and the frequency is high. In the numerical example in Sect. 17.5, the solutions are computed after the systems are preconditioned by a damped Helmholtz preconditioner [2].

17.3 The Noise Control Problem

17.3.1 Anti-noise Actuator Signal Optimization

The noise control problem is next presented briefly. A more detailed description is given in an earlier paper [1]. The problem is considered in the frequency domain, i.e. noise control is considered for one frequency at once; it should, however, be noted that the noise is not restricted to a single-frequency sound. The acoustic model is considered in an enclosed stochastic domain $\Omega(\mathbf{r})$, where $\mathbf{r}$ is a random vector that conforms to a known probability distribution $F(\mathbf{r})$. The sound pressure $p(\omega, \mathbf{x}, \mathbf{r}, \boldsymbol{\gamma})$ at an angular frequency ω is the sum of the sound pressures caused by noise and n anti-noise sources

$$p(\omega, \mathbf{x}, \mathbf{r}, \boldsymbol{\gamma}) = p_0(\omega, \mathbf{x}, \mathbf{r}) + \sum_{j=1}^{n} \gamma_j p_j(\omega, \mathbf{x}, \mathbf{r}), \tag{17.3}$$

where the pressure amplitude p_0 is due to the noise source, p_j is due to the jth anti-noise source, and γ_j is a complex coefficient defining the amplitude and phase

of the jth anti-noise source. The noise and anti-noise sources are located on the boundaries of Ω. The anti-noise defined by the coefficients γ_j is optimized so that the noise is minimized in a subdomain denoted by $\Xi(\mathbf{r}) \subset \Omega(\mathbf{r})$. For this, a noise measure is defined as

$$\begin{aligned} N(\omega, \mathbf{r}, \boldsymbol{\gamma}) &= \int_{\Xi(\mathbf{r})} \left| p(\omega, \mathbf{x}, \mathbf{r}, \boldsymbol{\gamma}) \right|^2 g(\mathbf{x}) d\mathbf{x} \\ &= \int_{\Xi(\mathbf{r})} p(\omega, \mathbf{x}, \mathbf{r}, \boldsymbol{\gamma}) \bar{p}(\omega, \mathbf{x}, \mathbf{r}, \boldsymbol{\gamma}) g(\mathbf{x}) d\mathbf{x}, \end{aligned} \tag{17.4}$$

where $g(\mathbf{x})$ is a weighting function and $\bar{p}$ is the complex conjugate of p. The expected value of the noise measure in the stochastic domain Ω is given by

$$E\big(N(\omega, \mathbf{r}, \boldsymbol{\gamma})\big) = \int N(\omega, \mathbf{r}, \boldsymbol{\gamma}) F(\mathbf{r}) d\mathbf{r}, \tag{17.5}$$

where $F(\mathbf{r})$ is the probability distribution of $\mathbf{r}$.

The objective function J for optimization of the noise control problem for the single frequency ω is chosen to be an approximation of the integral (17.5) and it is given by the numerical quadrature

$$J(\omega, \boldsymbol{\gamma}) = \sum_{j=1}^{m} w_j N(\omega, \mathbf{r}_j, \boldsymbol{\gamma}) F(\mathbf{r}_j), \tag{17.6}$$

where the pairs $(\mathbf{r}_j, w_j)$ give the quadrature points and weights. The optimization problem is defined as

$$\min_{\boldsymbol{\gamma} \in \boldsymbol{\Gamma}} J(\omega, \boldsymbol{\gamma}), \tag{17.7}$$

where $\boldsymbol{\Gamma}$ is the set of feasible controls, which, for simplicity, is here $\boldsymbol{\Gamma} = \mathbb{C}^n$. The optimal complex coefficients γ_i that give phases and amplitudes for anti-noise actuators are now given by the optimality condition $\nabla_{\boldsymbol{\gamma}} J = \mathbf{0}$, which leads to a system of linear equations.

17.3.2 Anti-noise Actuator Configuration Quality Measure

The actual configuration of anti-noise actuators, i.e. their number, locations and other properties such as size, determine the performance that can be obtained for a local noise control system.

Let us first define another noise measure function

$$\tilde{N}(\omega, \mathbf{a}, \mathbf{r}, \boldsymbol{\gamma}) = \int_{\Xi(\mathbf{r})} \left| p(\omega, \mathbf{a}, \mathbf{x}, \mathbf{r}, \boldsymbol{\gamma}) \right| g(\mathbf{x})\, d\mathbf{x}, \tag{17.8}$$

where $\mathbf{a} = (x_1, y_1, r_1, \ldots, x_n, y_n, r_n)$ is the anti-noise actuator configuration vector with (x_i, y_i) determining the location and r_i the radius of the ith anti-noise actuator. Noise reduction for the frequency ω in dB is now

$$R(\omega) = 10\log_{10}\frac{E(\tilde{N}(\omega, \mathbf{r}, \boldsymbol{\gamma}_{opt}))}{E(\tilde{N}(\omega, \mathbf{r}, \mathbf{0}))} = 10\log_{10}\frac{\int \tilde{N}(\omega, \mathbf{r}, \boldsymbol{\gamma}_{opt})F(\mathbf{r})\,d\mathbf{r}}{\int \tilde{N}(\omega, \mathbf{r}, \mathbf{0})F(\mathbf{r})\,d\mathbf{r}}, \quad (17.9)$$

for $\boldsymbol{\gamma}_{opt}$ which is optimized according to (17.7). The quality measure of anti-noise actuator configuration at the frequency ω is obtained by replacing integrals in (17.9) with trapezoidal quadratures

$$Q(\mathbf{a}, \omega) = 10\log_{10}\frac{\sum_j w_j^{\mathbf{r}}\tilde{N}(\omega, \mathbf{a}, \mathbf{r}_j, \boldsymbol{\gamma}_{opt})F(\mathbf{r}_j)}{\sum_j w_j^{\mathbf{r}}\tilde{N}(\omega, \mathbf{r}_j, \mathbf{a}, \mathbf{0})F(\mathbf{r}_j)}, \quad (17.10)$$

where $w_j^{\mathbf{r}}$ is quadrature weight from the trapezoidal rule for the integral of the probability distribution function F, and $\mathbf{r}_j$ is the co-ordinate triplet of the jth quadrature point.

17.3.3 Numerical Integration over Actuator

Circle-shaped anti-noise actuators are placed on a subdomain of a boundary surface, which is denoted by $A \subset \partial\Omega$. The subdomain A is composed of subdomains A_i such that $A = \bigcup_i A_i$. In order to allow convenient implementation of anti-noise actuator configuration optimization, a geometrical linear mapping is defined from the two-dimensional rectangular plane-domain $\tilde{A} = \bigcup_i \tilde{A}_i$ to A, such that the subdomains $\tilde{A}_i$ are mapped to A_i, respectively. Integrals are approximated by using a finite element solution on a triangular mesh. In order to improve integration accuracy of the boundary line, the triangles that reside on the anti-noise actuator boundary are divided into smaller triangles.

17.3.4 Noise Control in a Car Interior

As an example application of the method, noise control in a BMW 330i car interior is considered, see Fig. 17.1(a). The interior of the car excluding the driver is the domain $\Omega(\mathbf{r})$. The objective of the noise control is to minimize the noise in the driver's ears. Thus, Ξ is defined as a set

$$\Xi(\mathbf{r}) = \{\mathbf{e}_l, \mathbf{e}_r\} \subset \Omega(\mathbf{r}), \quad (17.11)$$

where $\mathbf{e}_l(\mathbf{r})$ and $\mathbf{e}_r(\mathbf{r})$ are the co-ordinates of the left and right ear, respectively. The noise measures (17.4) and (17.8) have now expressions

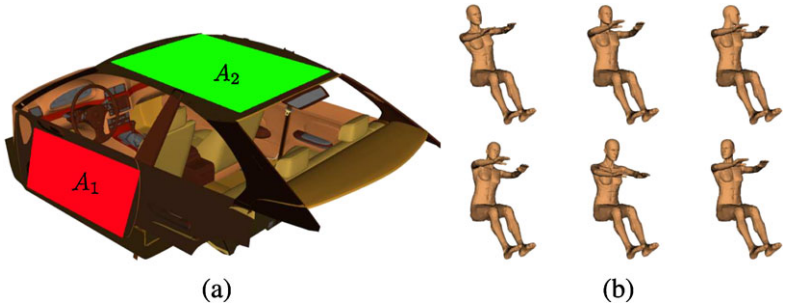


Fig. 17.1 (**a**) A three-dimensional model of a BMW 330i car interior. The subdomains A_1 and A_2 are marked with *red* and *green* colors, respectively. The subdomain A_3 is located on the right front door, which is not visible. (**b**) The driver's posture parameters from *left* to *right*: r_1 is the driver's sideways bending, r_2 is the forward bending, r_3 is head rotation. In the *upper* figures, the parameter's lowest value is shown and in the *lower* figures, the highest value is shown

$$\begin{aligned} N(\omega, \mathbf{r}, \boldsymbol{\gamma}) &= \left|p(\omega, \mathbf{e}_l, \mathbf{r}, \boldsymbol{\gamma})\right|^2 + \left|p(\omega, \mathbf{e}_r, \mathbf{r}, \boldsymbol{\gamma})\right|^2, \\ \tilde{N}(\omega, \mathbf{a}, \mathbf{r}, \boldsymbol{\gamma}) &= \left|p(\omega, \mathbf{a}, \mathbf{e}_l, \mathbf{r}, \boldsymbol{\gamma})\right| + \left|p(\omega, \mathbf{a}, \mathbf{e}_r, \mathbf{r}, \boldsymbol{\gamma})\right|. \end{aligned} \tag{17.12}$$

It is assumed that there is only the driver and no other passengers or significant objects in the car that would influence the sound propagation. The driver's variable properties like shape and posture are taken into account by considering a stochastic domain in the computation.

The driver is modeled by using the freely available Animorph library, based on [4]. Three parameters to model the driver are considered: r_1 is the driver's sideways bending angle, r_2 is the forward bending angle, and r_3 is the head rotation angle to left/right. These parameters are illustrated in Fig. 17.1(b) and their discrete values are as follows: $r_1 \in \{-20, -10, 0, 10, 20\}$, $r_2 \in \{-5, 0, 5, 10, 15\}$, and $r_3 \in \{-50, -25, 0, 25, 50\}$. The random variable vector $\mathbf{r} = (r_1, r_2, r_3)$ determines the posture of the driver.

In the car cabin interior, the noise source is modeled as a uniformly vibrating surface behind the leg room, which is a simplification of the real noise source. There are three possible surfaces where actuators may be located: on the left front door below window (A_1), on the roof (A_2) and on the right front door below window (A_3), see Fig. 17.1(a). The size of the door subdomains $A_{1,3}$ is 0.35×0.8 m^2 and the roof subdomain 1.0×1.0 m^2. These subdomains are placed and scaled beside each other so that they form a unit square, which makes it possible to use generic optimization formulation where optimization variables take values between $[0, 1]$. If an actuator crosses the boundary of the subdomain that it belongs to, it is cut so that only the part inside the subdomain is considered as an actuator. The anti-noise actuators are let to overlap freely and it also appears that they overlap in many optimized solutions. Overlapping could be avoided by penalizing such solutions during the optimization process.

To solve the Helmholtz equation (17.1) with the finite element method, a collection of meshes consisting of linear tetrahedra and triangles were generated with

Ansys ICEM CFD. Each mesh corresponds to a different driver posture and they were generated so that there are at least 10 nodes per wavelength at a 1000 Hz wave. The total number of meshes is $5^3 = 125$ which is the number of the parameter combinations (r_1, r_2, r_3).

The study was done in the frequency range 50–500 Hz with 25 Hz steps. This means that 18 frequencies were sampled. By employing the reciprocity principle, a sound source was placed in an ear. The acoustic model was solved for all 125 sampled driver's postures for both ears. Thus, discrete Helmholtz equations were solved $125 \times 18 \times 2 = 4500$ times for the optimal anti-noise control.

17.4 Evolutionary Multi-objective Optimization

A general form of a multi-objective minimization problem is

$$\begin{aligned} &\text{minimize} \quad \left\{ f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}) \right\} \\ &\text{subject to} \quad \mathbf{x} \in S, \end{aligned} \tag{17.13}$$

where $f_{1,\ldots,k} : \mathbb{R}^n \to \mathbb{R}$ are conflicting objective functions that are minimized by altering values of the design variables forming a vector $\mathbf{x} \in \mathbb{R}^n$ within a feasible region $S \subset \mathbb{R}^n$. The solution A is said to dominate the solution B if all components of $f(A)$ are at least as good as those of components of $f(B)$, with at least one strictly better component. Furthermore, A is non-dominated if it is not dominated by any feasible solution. Correspondingly, the solution A belongs to the Pareto optimal set if it is not dominated by any other feasible solution.

The multi-objective optimization problem for the locations and sizes of anti-noise actuators is defined to maximize the average expected attenuation obtained by local noise control at two frequency ranges simultaneously. The frequency ranges are given by the vectors $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_{n_\omega})$, $\boldsymbol{\iota} = (\iota_1, \ldots, \iota_{n_\iota})$. The objective functions are as follows:

$$f_1(\mathbf{x}) = \frac{1}{n_\omega} \sum_i Q(\mathbf{x}, \omega_i) \quad \text{and} \quad f_2(\mathbf{x}) = \frac{1}{n_\iota} \sum_i Q(\mathbf{x}, \iota_i), \tag{17.14}$$

where Q is the quality measure (17.10) and $\mathbf{x}$ is the design vector containing the location co-ordinates and radii of the anti-noise actuators.

Evolutionary multi-objective optimization algorithms (EMOA) (see, e.g., [6]) are among the widely used approaches in solving demanding engineering problems with multiple objectives. Different EMOAs employ various methods in the way they generate trial points and how they bring about the evolution of the population. Usually fitness is based primarily on dominance (non-dominated solutions are preferred), and secondarily on diversity (solutions on crowded regions are pruned).

Probably the most often referred and widely utilized algorithm in the above-mentioned category is the elitist non-dominated sorting genetic algorithm (NSGA-II) [8]. Yet, NSGA-II is claimed to have certain defects both in its performance

and in its basic foundations, and for this reason both NSGA-II and one algorithm that should overcome its defects, namely unrestricted population-size EMOA (UPS-EMOA) [3], are considered. In the following subsections short descriptions of both of these algorithms are given together with the corresponding parameters that are used in the numerical examples.

17.4.1 NSGA-II

Functioning of the dominance and diversity preservation based NSGA-II algorithm as it is implemented here is described briefly as follows:

Step 1. Create an initial (parent) population of $n_{beginpop} = 10n_{pop} = 10000$ population members randomly.

Step 2. Evaluate objective function values for the initial population and choose the $n_{pop} = 1000$ best ones based on non-domination.

Step 3. Generate n_{pop} trial points to create a child population by using the simulated binary cross-over operation (SBX, see [7]). The following parameters have been used: cross-over probability $p_c = 0.9$, mutation probability $p_m = 1/n_{vars}$ where n_{vars} is the number of variables, SBX cross-over parameters for the crossover $\eta_c = 10$, and mutation $\eta_m = 10$, tournament size $n_{tour} = 2$.

Step 4. Evaluate objective function values for the child population.

Step 5. Combine the parent and the child populations. Identify non-dominated solutions from the combined population. Create the next parent population by taking solutions from the combined population to the new one:

   a. If there is excess of non-dominated solutions to fit into the next population, prune such excess solutions which are located in more crowded areas (diversity preservation).
   b. If there are not enough non-dominated solutions to fill the next population, identify again non-dominated solutions remaining in the combined population, and continue this cycle until the population is filled.

Step 6. If the number of allowed generations is not exceeded, or the budget for objective function evaluations is not exhausted, go back to Step 3.

Unfortunately, it seems not to be widely fathomed that this type of algorithm suffers from several theoretical drawbacks, such as oscillation [3] (lack of convergence [10]), deterioration of the population, and lack of performance.

It is said that the method involves oscillation if a solution close to the Pareto optimal set is replaced by another non-dominated solution which improves diversity but is at the same time located much farther from the Pareto optimal set. If in the history of all the evaluated solutions there exist solutions that dominate the solutions in the current population, then the population is said to be deteriorated. If deterioration occurs, it suggests that the algorithm has wasted some objective function evaluations,

and could have actually performed better. This behaviour also contributes to general lack of performance.

17.4.2 UPS-EMOA

The basic feature of the recently published UPS-EMOA [3] is the use of a population which has no artificial size limit. Instead, the population always contains all the non-dominated solutions found during the optimization process, and thus the population expands. Theoretically, this may lead to a situation where storage requirements are unbounded. In practice, we have not witnessed such behaviour, as the number of evaluations is kept finite. By expanding population, the algorithm overcomes some problems of the current EMO approaches, such as oscillation (lack of convergence), deterioration of the population, and lack of performance. Steps of the UPS-EMOA implementation used in this paper are presented as follows:

Step 1. Initialize the population within the given search space using $n_{beginpop} = 10000$ points covering the space as uniformly as possible. Points are created using a space-filling Hammersley sequence [9].
Step 2. Evaluate the objective function values of the new points.
Step 3. Combine the current population with the new points. Identify non-dominated solutions, and move all these to the next population. If the minimum size of the population $n_{min} = 50$ is not reached, take non-dominated solutions from the remaining points, and continue until the minimum size is reached.
Step 4. Select randomly $n_{burst} = 260$ points from the current population to be used as parents. Generate one new child point for every parent point using the point generation mechanism of differential evolution (DE, see [12]), using cross-over probability $C_r = 0.5$ and the scaling factor $F = 1.0$. In the creation of the new point, all points in the current population may participate. Points which are not inside the given search space are truncated to the border, similarly as in NSGA-II.
Step 5. Evaluate the objective function values of the child population, and if the budget for objective function evaluations is not exhausted, go back to Step 3.

17.4.3 Comparison of EMOAs by Hypervolume Measure

With multi-objective optimization algorithms that produce an approximation of the Pareto-optimal set, measuring the performance of a given algorithm is far from trivial. To characterize the goodness of the solution set, all solutions should be as close as possible to the real Pareto optimal set (*closeness*) and the solutions should cover the whole Pareto optimal set as well as possible (*diversity*), meaning that the distribution of the solutions along the Pareto optimal set should be even, and the extent of the solutions should be as high as possible.

Recently, a hypervolume indicator [17] has gained popularity both as a performance metric and as a selection criterion in EMOAs. The hypervolume defines the volume of the objective space dominated by the given solution set, and as such it can give information about both closeness and diversity at the same time.

In this study, hypervolume is used as a performance metric to make a rough comparison between two selected algorithms.

17.5 Numerical Experiments

Four numerical optimization test cases are considered to demonstrate and analyze the efficiency of the method. All test cases involve local noise control in a car interior (see Fig. 17.1) as explained in Sect. 17.3.4. The test cases are as follows:

- Test case #1: 2 fixed-size actuators (4 design variables)
- Test case #2: 3 fixed-size actuators (6 design variables)
- Test case #3: 3 variable-size actuators (9 design variables)
- Test case #4: 8 variable-size actuators (24 design variables)

Equations (17.14) are considered as contradicting objective functions, with two frequency ranges 50–275 Hz and 275–500 Hz corresponding to vectors $\boldsymbol{\omega} = [50, 75, \ldots, 250]$ and $\iota = [275, 300, \ldots, 500]$. For the test case #1, the design vector $\mathbf{x} = (x_1, x_2, r, x_3, x_4, r)$, where $r = 0.112$ m, i.e. there are four design variables. For the test case #2, similarly $\mathbf{x} = (x_1, x_2, r, x_3, x_4, r, x_5, x_6, r)$. For the test cases #3 and #4, where the actuator radius $r \in [0.05, 0.175]$ m is also a design variable, $\mathbf{x} = (x_1, \ldots, x_n)$, with $n = 9$ and $n = 24$, respectively. The test cases #1–#3 were run until the limit of 100000 objective function evaluations and the test case #4 was run until the limit of 200000 objective function evaluations.

The first test case was chosen in order to present a simple case with a low number of design variables. The test cases #2–#4 present more difficult optimization problems, where EMO approaches may not be able to find a global unambiguous optimum, which is a well-known feature of the used methods when the search space is large due to the number of design variables and when there are plenty of local minima in the problem. Nevertheless, these methods are able to bring about a significant improvement, when compared with a sophisticated engineer guess.

17.5.1 Convergence of Multi-objective Optimization Methods

To justify the choice of using UPS-EMOA as a preferred optimization algorithm for the presented problem, the convergence was compared to the NSGA-II by evaluating hypervolumes of the solution fronts (see Sect. 17.4.3). The hypervolume as a function of the number of objective function evaluations is plotted for all test cases in Fig. 17.2.

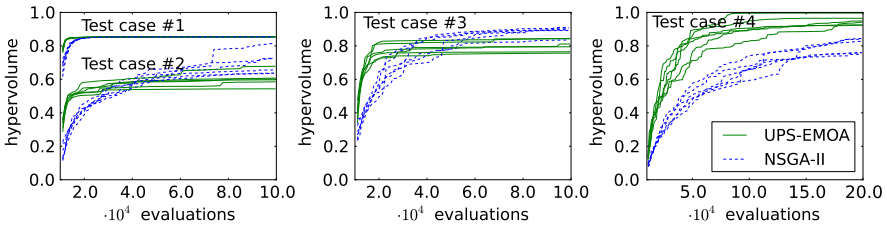


Fig. 17.2 Hypervolume as a function of the number of objective function evaluations with UPS-EMOA and NSGA-II. Six random number generator seed numbers for each test case and algorithm. The resulting lines of different test cases should not be compared to each other, due to incompatible scales

For the test case #1, UPS-EMOA converges notably faster to its maximum, already at 14000 evaluations, while NSGA-II reaches the same level at 20000 evaluations. This is the only test case where robust convergence towards the identical solution front is obtained and it is due to the low number of design variables, $n_{vars} = 4$. For all test cases, it is clearly seen that UPS-EMOA convergence is notably faster in the beginning of the optimization process. However, none of the test cases #2–#4 converge robustly towards a single solution front, which is due to the larger search space with plenty of local minima.

For the test cases #2 and #3, NSGA-II eventually finds better solution fronts, despite its slower convergence in the beginning. In Fig. 17.5 (on p. 333), the final solution fronts after 99440 objective function evaluations are plotted, where it can clearly be seen that while the right part of the front is identical, on the left part the NSGA-II has progressed further. We suggest that this is due to concentrated point density of the UPS-EMOA results on the right part of the front, leading to a situation where points on the left have only diminishing probability to be selected as parents. Thus the development of the front in that region suffers.

For the test case #4, where there are 24 design variables and for all runs, UPS-EMOA converges faster and gives better solution fronts than NSGA-II. As a conclusion, UPS-EMOA is clearly a better choice, when the CPU time usage is limited and/or when a larger number of design variables is involved.

17.5.2 Example Solutions

In Fig. 17.3, the solution fronts for all test cases are shown, obtained by UPS-EMOA. These fronts can be compared to the objective function values obtained by sophisticated engineer guesses (see Fig. 17.4) that are plotted as well. It is clearly seen that optimization improves the noise control remarkably. The figure also illustrates the big improvement obtained when the number of anti-noise actuators is increased; compare the solution fronts for the test cases #1 to #2 where the number of actuators increases from 2 to 3 ($\sim$ 3–5 dB improvement in both objective function values), and the solution fronts for the test cases #3 to #4, where the number

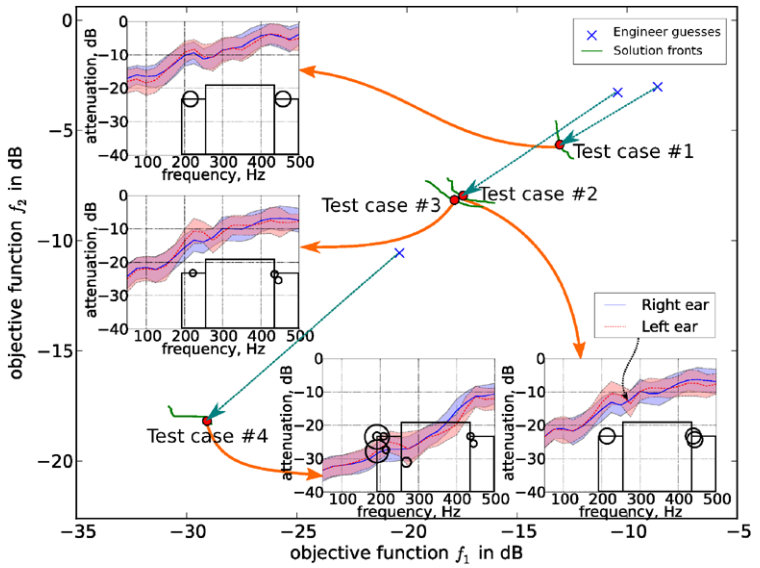


Fig. 17.3 The solution fronts for all test cases obtained by UPS-EMOA. *Blue* crosses correspond to non-optimized engineer guesses, shown in Fig. 17.4. One solution from each front is selected and shown in small subfigures, similarly as in Fig. 17.4

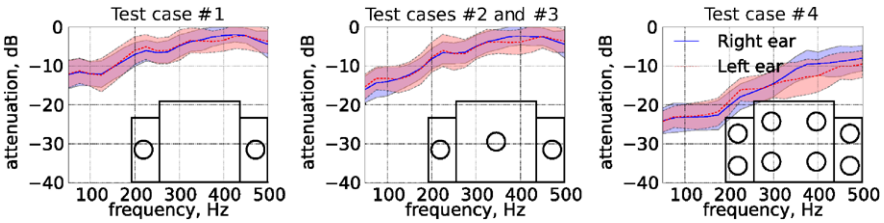


Fig. 17.4 Engineer guesses of good anti-noise actuator configurations. The figures show (1) the anti-noise actuator configuration in the subdomains A_1, A_2, and A_3 (see Fig. 17.1), and (2) the expected value of attenuation in the left and right ear with standard deviation (the shaded region). Corresponding objective function values of (17.10) are plotted in Fig. 17.3

of actuators increases from 3 to 8 (~ 10 dB improvement in both objective function values).

In Fig. 17.5, the solution fronts for the test cases #2 and #3 are given after 99440 objective function evaluations. Three solutions are selected for both test cases from a single front obtained by UPS-EMOA. Both test cases have three anti-noise actuators, but the difference between them is that while in the test case #2, the sizes (radii) of each actuator are constant, in the test case #3 they may vary. It is seen that this increase in the degree of freedom gives only a 0.2–0.5 dB enhancement in objective function values. It is also seen that smaller anti-noise actuators seem to be

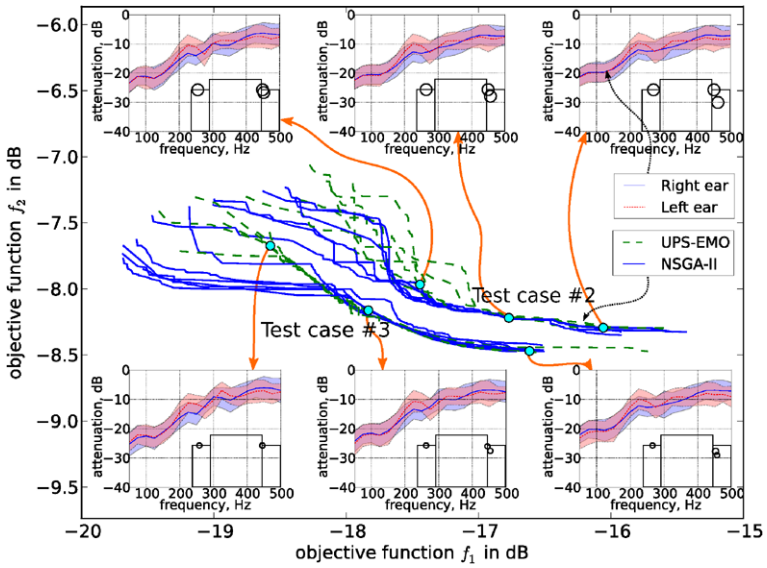


Fig. 17.5 The solution fronts for the test problems #2 and #3 obtained by NSGA-II and UPS-E-MOA, six random number generator seed numbers. Three solutions (*cyan circles*) from a single UPS-EMO front are selected for both test cases and they are shown in small subfigures, similarly as in Fig. 17.4

preferable in this case. Similar behavior in actuator placements is seen in both test cases.

17.6 Conclusions

A novel technique was proposed to find optimal locations for anti-noise actuators by using a finite element model based numerical evaluation method for optimal local noise control. The optimization of anti-noise actuator configuration, i.e. the placing and size of each actuator, was formulated as a multi-objective optimization problem so that optimal noise reduction at two frequency ranges could be obtained.

As an example problem, local noise control in a car interior with a driver in varying postures was considered and numerical results were presented. Two evolutionary multi-objective algorithms, UPS-EMOA and NSGA-II were used as optimization methods and their performance was compared. It was found that in all test cases UPS-EMOA was converging faster in the beginning of the optimization process, but NSGA-II was able to give better final solution fronts in two test cases.

Numerical examples clearly demonstrated that, by employing optimization of anti-noise actuator configuration, it is possible to obtain a significant improvement in the objective function values over sophisticated engineer guesses.

Acknowledgements The research was funded by Academy of Finland, the grant #250979.

References

1. Airaksinen T, Heikkola E, Toivanen J (2011) Local control of sound in stochastic domains based on finite element models. J Comput Acoust 19(2):205–219
2. Airaksinen T, Pennanen A, Toivanen J (2009) A damping preconditioner for time-harmonic wave equations in fluid and elastic material. J Comput Phys 228(5):1466–1479
3. Aittokoski T, Miettinen K (2010) Efficient evolutionary approach to approximate the Pareto-optimal set in multiobjective optimization, UPS-EMOA. Optim Methods Softw 25(4–6):841–858
4. Bastioni M, Re S, Misra S (2008) Ideas and methods for modeling 3d human figures. In: Shyamasundar RK (ed) Proceedings of the 1st Bangalore annual compute conference. ACM, New York. doi:10.1145/1341771.1341782
5. Bermúdez A, Gamallo P, Rodríguez R (2004) Finite element methods in local active control of sound. SIAM J Control Optim 43(2):437–465
6. Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, Chichester
7. Deb K, Agrawal RB (1995) Simulated binary crossover for continuous search space. Complex Syst 9(2):115–148
8. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
9. Hammersley JM (1960) Monte Carlo methods for solving multivariable problems. Ann NY Acad Sci 86:844–874
10. Laumanns M, Thiele L, Deb K, Zitzler E (2002) Combining convergence and diversity in evolutionary multiobjective optimization. Evol Comput 10(3):263–282
11. Nelson PA, Elliott SJ (1993) Active control of sound. Academic Press, London
12. Price KV, Storn RM, Lampinen JA (2005) Differential evolution. A practical approach to global optimization. Springer, Berlin
13. Provatidis CG, Mouzakitis ST, Charalampopoulos GN (2009) Simulation of active noise control in enclosures using direct sound field prediction. J Comput Acoust 17(1):83–107
14. Shorter P (2008) Recent advances in automotive interior noise prediction. SAE technical paper 2008-36-0592. doi:10.4271/2008-36-0592
15. Stanef DA, Hansen CH, Morgans RC (2004) Active control analysis of mining vehicle cabin noise using finite element modelling. J Sound Vib 277(1–2):277–297
16. Thompson LL (2006) A review of finite-element methods for time-harmonic acoustics. J Acoust Soc Am 119(3):1315–1330
17. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study. In: Proceedings of the 5th international conference on parallel problem solving from nature (PPSN V). Springer, London, pp 292–301

Chapter 18
From the Idea of Bone Remodelling Simulation to Parallel Structural Optimization

Michal Nowak

Abstract The paper provides an overview of the structural optimization system development. The basis and also the primary idea for algorithm formulation was the bone remodelling phenomenon leading to the optimization of the trabecular net within the bone. The idea was completed with theorems concerning the surface constant strain energy principle to form the biomimetic optimization system. The paper describes the key element of the optimization procedure: our own mesh generator called Cosmoprojector. It also presents the concept of Finite Element mesh parallel generation as well as Finite Element Analysis in a parallel environment as a recent enhancement of the presented method. Finally, it presents some results of computations obtained with the use of biomimetic structural optimization.

18.1 Introduction

The Wolff law, stated in the 19th century, says that bone is capable of adapting itself to mechanical stimulation. After carrying out many experiments, it is now clear that the number and organization of beams in trabecular bone tend to a mechanical optimum. There are many models of bone remodelling [2, 3, 7, 9] used for the adaptation simulations of bone, treated as a continuum material. The main idea behind that is to prepare a model for bone adaptation as a material of specific properties. These properties vary and depend on the load history. The progress in computer hardware technology and parallel computations now enable modelling of the bone adaptation process using the real topology of the trabecular bone with the use of a linear model of the trabecula [3, 7]. The latter one is justified by experimental investigations stating that on the trabecular level bone can be treated as a linear material. Such an approach can be considered as very useful, especially when the details of mechanical stimuli are discussed. The trabecular bone mechanical adaptation process is similar to all other structural optimization problems. In this context, the bone remodelling could be also the model of the structural optimization procedure.

M. Nowak (✉)
Department of Machine Design Method, Poznan University of Technology, ul. Piotrowo 3, 60-965 Poznan, Poland
e-mail: Michal.Nowak@put.poznan.pl

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_18,

The paper presents the main stages of the numerical implementation of this concept, from the idea of bone remodelling simulation to parallel structural optimization. Section 18.2 covers the basic assumptions of numerical simulation of the trabecular bone remodelling process. Section 18.3 recalls the theoretical background for treating the biological process as a pattern for the structural optimization. Section 18.4 presents the practical realisation of the structural optimization method based on the bone remodelling model. Section 18.5 describes the recent enhancement of the presented system—parallelisation of the Finite Element Method (FEM) mesh generation. Section 18.6 presents the numerical example summarizing the overview of the system.

18.2 Simulations of Trabecular Bone Adaptation to Mechanical Loading

The healthy bone trabecular tissue has a very sophisticated structure. The tissue forms a network of beams called trabeculae and this structure is capable of handling a wide range of loads. The length of the trabecula amounts to one or two hundred micrometers whereas its diameter is about 50 micrometers. This structure is continually rebuilt so that the whole bone tissue is replaced in the course of about three years and the process is called the trabecular bone adaptation or remodelling. The examined phenomenon is based on the balance between bone resorption and formation of the new tissue. Thus the trabecular bone is capable of repairing the fractures, by simply replacing part of its structure with the new one.

The phenomenon of trabecular bone adaptation has two important attributes. Firstly, the mechanical stimulation is needed to conserve the rebuilding balance. Secondly, the process of resorption and formation occurs on the trabecular bone surface only. The need of effective simulation of bone remodelling was the beginning of the presented optimization system [4].

The developed generic three-dimensional system for bone remodeling simulation employing the Finite Element Method (FEM) consists of the following three blocks: FEM preprocessing, FEM solution, and optimization and modification procedures. The system used in this study uses the algorithm of bone remodeling stimulated by mechanical loading, based on the strain energy density (SED) distribution. The beams of trabecular bone are assumed to be made of isotropic linear elastic material where the marrow space is treated as voids. In contrast to other approaches used so far, the system mimics the real bone geometry evolution where not only the volumetric FEM mesh but also the surface of the trabecular network is controlled during the simulation. Because the remodelling process occurs only on the surface of the trabecular bone, only the ‘surface’ layer of the structure is taken into consideration during the simulation process. In contrast to other voxel models, the approach adopted in this system does not relay on earlier voxel discretisations, but mimics the natural evolution of the bone tissue as the biological process of bone formation and resorption.

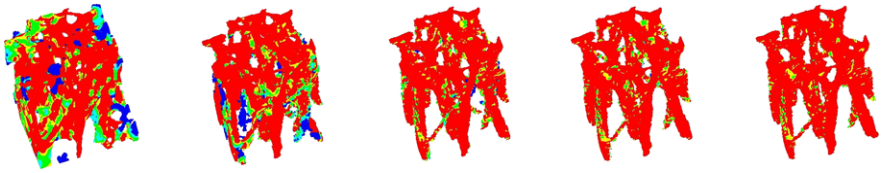


Fig. 18.1 Remodelling simulation of the trabecular bone sample under compression with the use of the presented system [4]

Adaptation to the mechanical stimulation results in adaptation of the surface position in the virtual space. The surface position adaptation is realized on the two-dimensional input images in the graphical form, by adding or removing pixels. Thus, both the consolidation and the separation of the tissue can be modeled easily. Figure 18.1 shows the remodeling simulation of the trabecular bone sample under compression with the use of the presented system [4].

18.3 Remodelling as a Structural Optimization Process

The two attributes of trabecular bone remodelling, mechanosensitivity and surface adaptation, can be described from the point of view of mechanics. Bearing in mind the design with optimal stiffness [6, 10], one can conclude that for the stiffest design the strain energy density along the shape to be designed must be constant:

$$u_{\epsilon_s} = \text{const.} \tag{18.1}$$

In the case of bone, the remodelling scenario described above, based on the phenomenological model, seems to realize the postulate of the constant value of the strain energy density. By balancing the SED value on the bone surface, the stiffness of the entire structure is ensured. In ideal conditions when the bone structure is only rebuilt, also the volume constraint is valid. In the real process, the bone volume undergoes continuous change. In this study the presented examples employ the scenario of bone remodelling which assumes changes of the bone volume in the successive iterations.

So, the fixed volume constraint, resulting from the minimum compliance discussion, is not a case in the bone remodelling. The optimization goal can also be formulated as a minimum volume problem with assumed fixed strain energy. The resulting condition concerning the strain energy density is the same as in the case of the minimum compliance, so the value of the strain energy density on the designed surface must be equal when the volume is minimal by the assumed value of the strain energy in the structure.

In the model of bone remodelling, there is a special value of strain energy density—the energy of homeostasis where the balance between resorption and formation of the bone tissue is perfect. Figure 18.2 shows the computation scheme with strain energy density as a remodelling criterion.

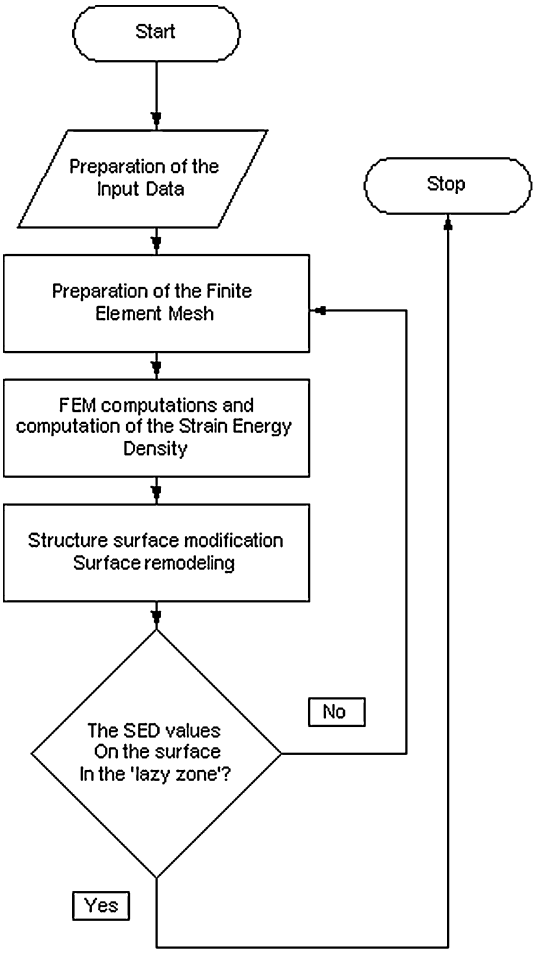


Fig. 18.2 Computation scheme with strain energy density (surface remodelling) as a remodelling criterion

Now, after the discussion of the stiffest design issues, the numerical implementation of the bone remodelling can be treated as a par excellence structural optimization procedure.

A more detailed description of the assumptions and the arguments can be found in [5, 6, 10].

18.4 Structural Optimization Method Based on the Bone Remodelling Scheme

The developed numerical trabecular bone remodelling simulation environment was dedicated to and tested for biomechanical purposes. To compare the optimization procedure based on trabecular bone surface adaptation to the standard optimiza-

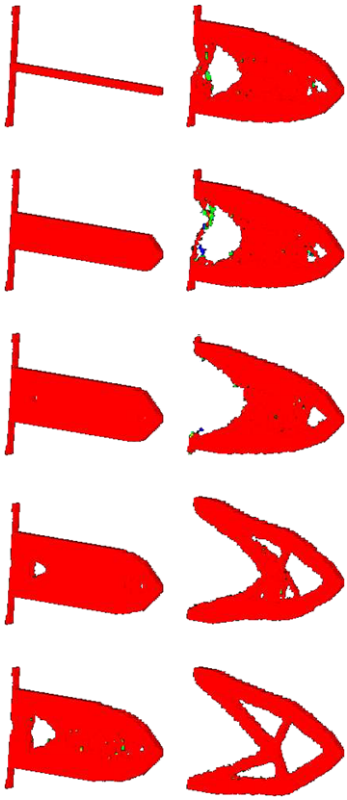


Fig. 18.3 Optimization results of the cantilever beam using the presented system, based on the trabecular bone surface adaptation—an empty domain [5]

tion method, a typical topology optimization method example, i.e. the bending cantilever beam, was chosen. To illustrate the special features of the proposed optimization technique, the domain was reduced to a minimum, and the fixed part of the structure was simply connected to the loaded part. As it was possible to add material during the simulation, the result shown in Fig. 18.3 is very similar to the three-dimensional solution given in the Topology Optimization book by Bendsøe and Sigmund [1, p. 25]. The result is similar, but starting from such domain, it is not possible to achieve the solution using the standard method of topology optimization. This feature, which facilitates adaptation of the structure conserving functional configurations, necessary in the case of biological structures, can also be valuable for mechanical structures (in space or civil engineering) [5].

18.5 Parallellisation of the Mesh Generation Procedure

Due to necessity of surface control and the specific rules of structural adaptation, the optimization procedure requires relatively big computational effort to produce an appropriate and accurate mesh. The development of the structural optimization

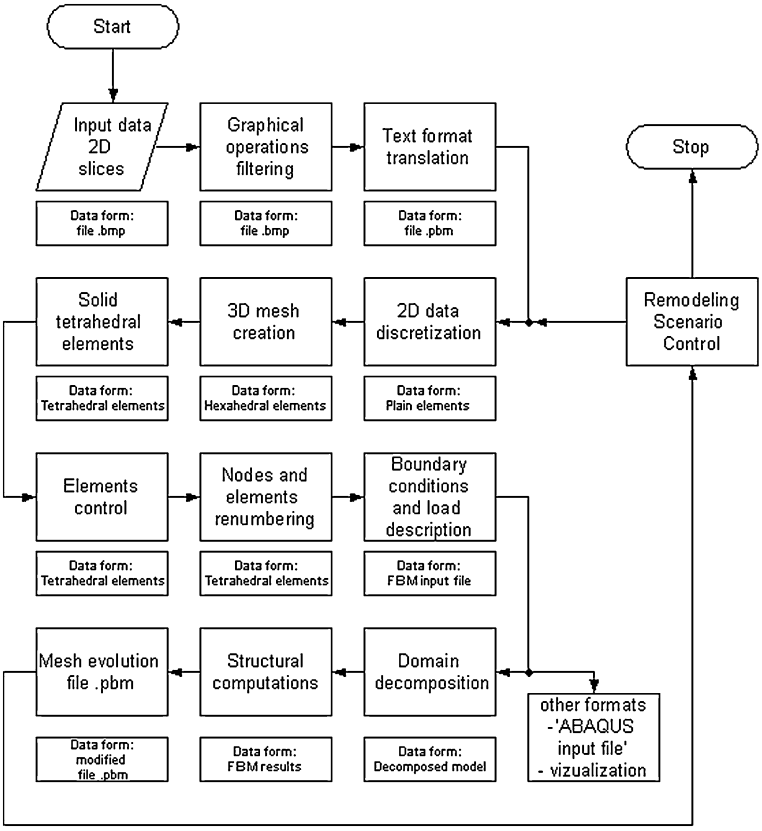


Fig. 18.4 Cosmoprojector—the detailed scheme of the mesh generation procedure

method was thus focused on the use of a parallel computational environment to generate the structural mesh.

The presented method is based on structure evolution. Thus, a bottleneck of the optimization process is finite element mesh generation for each step of structural evolution. The finite element mesh generator, Cosmoprojector, was originally dedicated to mesh creation for biological entities. Since the visualization for the biological entities is based on the digital images, e.g., Computer Tomography, the input to the system is also based on the collection of the two-dimensional images. After some graphical operations the images of slices are directly used for the building of the three-dimensional finite element mesh. The two-dimensional image is first translated into the bitmap where '0' represents void and '1' the tissue. On the bitmap the initial step of discretisation is executed. The aim of this first step is to describe the areas with tissue (or just the material in the case of optimization issues). The discretisation procedure produces a two-dimensional network of tetragonal elements, according to the tissue image shape. The discretised two-dimensional image is projected to the subsequent one. If there are areas containing material on both images,

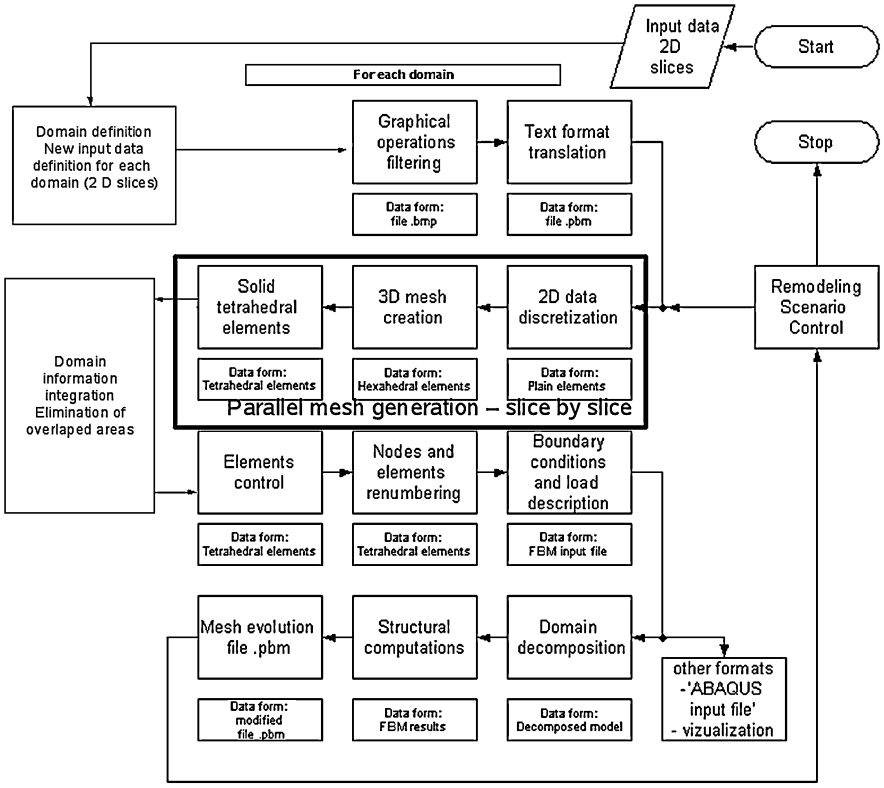


Fig. 18.5 Cosmoprojector—the parallel mesh generation procedure

the boxes are created. Each box is in turn translated into six tetrahedral volume elements. The information about nodes and elements is stored in a special database and translated into an ABAQUS finite element system input file. The system has to enable the surface control, so in the whole structure the elements on the structure are distinguished. The detailed scheme of the mesh generation procedure is shown in Fig. 18.4.

The strain energy density computations are carried out in a parallel environment, which is a condition to solve bigger problems. But the same question concerns mesh generation, especially if the mesh element's number is of order 10^6.

To increase the capabilities of the optimization system the mesh generation tool was parallelised. The scheme of the parallel mesh generation procedure is shown in Fig. 18.5. In the natural way the mesh generation for the whole domain can be divided into independent tasks. The only change is the modification of input data necessary to define the overlapping areas. The aim of overlapping areas is to ensure that the slice-by-slice mesh creation procedure is independent of the number of processors used in the computation. The parallel finite element mesh generator was successfully tested up to 200 nodes and is able to create meshes of order of millions tetrahedral elements.

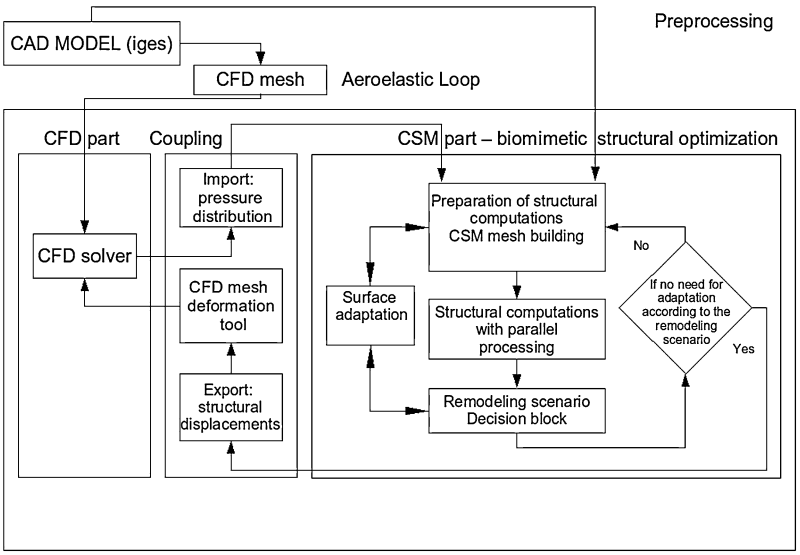


Fig. 18.6 The algorithm for aeroelastic analysis coupled with the biomimetic structural optimization

18.6 The Multiphisics Example—Biomimetic Structural Optimization Coupled with Aeroelastic Analysis

As an example of the biomimetic structural optimization method described here, the problem of internal wing structure design is presented. To design an aircraft structure, the coupled fluid-structure interactions (FSI) simulations are crucial. On the other hand, for the structural design optimization techniques have to be used.

There are many examples of using optimization techniques to design the structural elements of an aircraft [8]. In the recent years, especially the topology optimization method has been introduced to the designing processes. A good industrial example is the structure of the Airbus A380 wing. The structural elements of the wing were designed in two designing steps. First, the optimal material distribution was defined using the topology optimization—the SIMP method. Then, after extraction of geometry from the topology optimization results, the model for size and shape optimizations was derived. The size and shape optimizations were the next step in the wing designing process. Splitting the topology and then the size and shape optimizations is necessary because of completely different optimization methods used in each case. The biomimetic approach described here allows comprising the optimizations of size, shape, and topology. Also the varying loads during the aeroelastic analysis due to assumptions adapted directly from the trabecular bone remodelling phenomenon do not interfere with the optimization process.

Figure 18.6 depicts the algorithm for coupling aeroelastic analysis with structural optimization. The approach presented here is based on the assumption that different codes will be used separately for each part of the simulation field. The starting

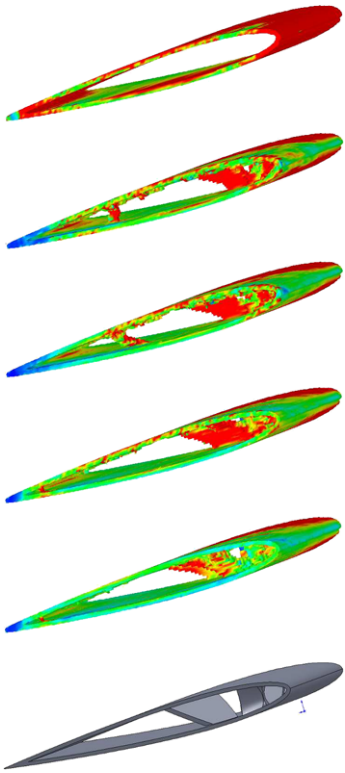


Fig. 18.7 The results of coupled aeroelastic and optimization procedures. From the top down: selected coupled simulation steps and the CAD model of the optimised structure

domain was an empty domain in the internal area of the airfoil. The outer shape of the wing retains its form during the whole simulation process. The optimization procedure starts by computing the aerodynamic load resulting from the CFD computations. Then, the pressure distribution on the outer wing surface is interpolated on the structural mesh and the optimization loop is performed. The results of coupled aeroelastic analysis and biomimetic optimization for the whole internal wing structure is shown in Fig. 18.7.

18.7 Conclusions

The idea of bone remodelling simulation was the foundation of the studies which resulted in the creation of the numerical system capable of mimicking a real biological phenomenon. This system was a numerical base for the structural optimization method based directly on the principle of constant strain energy density on the surface. The biomimetic structural optimization method has some unique properties. The domain independence, functional configurations during the process of optimization, and a possible solution of multiple load problems are special features which provide new possibilities in the area of structural optimization. Thus, using

the approach presented above it is possible to comprise optimizations of size, shape, and topology with no need to define parameters. The presented method is able to produce results similar to the standard method of topology optimization and can be useful in mechanical design, especially when functional structures are needed during the optimization process, broadening the spectrum of possible applications. Due to parallelisation of both the structural analysis of strain energy density distribution and volume mesh generation, the presented method can be useful in real industrial problems. The initial concept of the optimization system was dedicated to the bone remodelling purposes. Over time, the concept has evolved, and today it can be a useful tool for mechanical design.

Acknowledgements This work was partially supported by the Polish Ministry of Science and Higher Education under the grant no. N N518 328835.

References

1. Bendsøe MP, Sigmund O (2003) Topology optimisation. Theory, methods and applications. Springer, Berlin
2. Huiskes R, Ruimerman R, van Lenthe GH, Janssen JD (2000) Effects of mechanical forces on maintenance and adaptation of form in trabecular bone. Nature 405:704–706
3. Niebur GL, Feldstein MJ, Yuen JC, Chen TJ, Keaveny TM (2000) High-resolution finite element models with tissue strength asymmetry accurately predict failure of trabecular bone. J Biomech 33(12):1575–1583
4. Nowak M (2006) A generic 3-dimensional system to mimic trabecular bone surface adaptation. Comput Methods Biomech Biomed Eng 9(5):313–317
5. Nowak M (2006) Structural optimization system based on trabecular bone surface adaptation. Struct Multidiscip Optim 32(3):241–249
6. Pedersen P (2003) Optimal designs—structures and materials—problems and tools. Technical University of Denmark. Draft, ISBN 87-90416-06-6
7. Ruimerman R, Van Rietbergen B, Hilbers P, Huiskes R (2003) A 3-dimensional computer model to simulate trabecular bone metabolism. Biorheology 40(1–3):315–320
8. Schramm U, Zhou M (2006) Recent developments in the commercial implementation of topology optimization. In: IUTAM symposium on topological design optimization of structures, machines and materials. Springer, Berlin, pp 239–248
9. Tsubota K, Adachi T, Tomita Y (2002) Functional adaptation of cancellous bone in human proximal femur predicted by trabecular surface remodeling simulation toward uniform stress state. J Biomech 35(12):1541–1551
10. Wasiutynski Z (1960) On the congruency of the forming according to the minimum potential energy with that according to equal strength. Bull Acad Pol Sci, Ser Sci Tech 8(6):259–268

Part V
Mathematical Models Generated by Modern Technological Problems

Chapter 19
Uncertainties in Contact Mechanics and Shape Optimization Problems

Nikolay Banichuk and Svetlana Ivanova

Abstract Shape optimization problems with uncertainties are considered for a rigid punch interacting with an elastic medium. Considered loads are supposed to be given with incomplete data or described by random variables. Corresponding investigations are respectively performed in the framework of a minimax (worst case scenario) approach or a stochastic approach. As a result of the proposed approaches and applied analytical methods, the optimal designs of the punch are obtained and presented in the paper for various definitions of uncertainties.

19.1 Introduction

In structural optimization problems the loading scenario is usually assumed to be known. In particular, the regions of applications of the loads, the form of the distribution of the forces and their value are assumed to be specified [1, 6, 7]. No changes are permitted in the specification of the external loads and the optimum solutions determined under these conditions are sensitive to a change in the problem parameters. However, the situation is quite different in many applications: the regions where the load act, the distribution of the applied forces and their limiting values are indeterminate and depend on a number of random quantities. To describe the uncertainties that arise and to formulate optimization problems with incomplete information, different approaches can be employed [5], for example, a probability approach [4, 5] based on a specification of the probability density function of the random loads. The limited nature of this approach is the fact that in many applications this function is unknown. There is another approach which does not use the probability density function; this is a guaranteed approach based on a minimax (worst case scenario) description and on a determination of the unknown parameters in a calculation of the "worst" case. In this approach the applied "indeterminate" loads do not need to

N. Banichuk (✉) · S. Ivanova
Ishlinsky Institute for Problems in Mechanics, Russian Academy of Sciences (RAS), Prospekt Vernadskogo 101, 119526 Moscow, Russia
e-mail: banichuk@ipmnet.ru

S. Ivanova
e-mail: ivanova@ipmnet.ru

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_19,

be described by a probability density distribution and *pertain* to a set of admissible loads. This approach *enables* a deterministic specification of the uncertainties to be used and enables modelling and optimization methods to be developed.

Below we present methods of optimizing the shape of a rigid punch under quasistatic conditions using the worst case scenario approach and the probabilistic approach separately.

19.2 Basic Relations and Elements of the Formulated Structural Optimization Problem

The interaction of a rigid punch with the elastic half-space $z \geq 0$ is considered in the rectangular coordinate system $0xyz$. The boundary of the elastic half-space Ω ($z = 0$) contains the contact domain Ω_f representing the base of the punch, the region Ω_0 which is free of loading q and the regions Ω_q^i, $i = 1, 2, \ldots, N$, of application of external forces q, i.e.

$$\Omega = \Omega_f + \Omega_0 + \Omega_q,$$

$$\Omega_q = \bigcup_{i=1}^{i=N} \Omega_q^i, \qquad \Omega_f \cap \Omega_0 = 0,$$

$$\Omega_f \cap \Omega_q^i = 0, \qquad \Omega_q^i \cap \Omega_0 = 0.$$

The surface of the punch penetrated into an elastic medium without friction is given by the equation

$$z = \begin{cases} f(x, y), & (x, y) \in \Omega_f, \\ 0, & (x, y) \in \partial\Omega_f, \end{cases}$$

where $f(x, y)$ a is positive, continuous, and smooth function. In what follows, this function will be considered as an unknown design variable.

External forces $q = \{q_x, q_y, q_z\}$ applied to the domains Ω_q are considered as uncertainty functions depending on incomplete data or random variables, i.e.

$$q \in \Lambda_q,$$

where Λ_q is a set described by some limits or random characteristics. Boundary conditions at Ω have the form

$$w = f(x, y), \qquad \sigma_{xz} = 0, \qquad \sigma_{yz} = 0, \qquad (x, y) \in \Omega_f, \tag{19.1}$$

$$\sigma_{zz} = q_z, \qquad \sigma_{xz} = q_x, \qquad \sigma_{yz} = q_y, \quad (x, y) \in \Omega_q, \tag{19.2}$$

$$\sigma_{zz} = 0, \qquad \sigma_{xz} = 0, \qquad \sigma_{yz} = 0, \quad (x, y) \in \Omega_0. \tag{19.3}$$

Here σ_{xz}, σ_{yz}, and σ_{zz} are components of the stress tensor, w is the projection of the displacement vector on the z-axis, and q_x, q_y, and q_z are given load distributions. The contact pressure distribution p can be found as a solution of the theory of the elasticity problem with the boundary conditions (19.1)–(19.3) for the normal stresses σ_{zz}, i.e.

$$p = -(\sigma_{zz})_{z=0}, \quad (x, y) \in \Omega_f.$$

Then the resulting force P applied to the punch and the total applied moments M_x, M_y (with respect to the axes x and y) can be estimated as

$$P = \int_{\Omega_f} p d\Omega_f, \qquad M_x = \int_{\Omega_f} y p d\Omega_f, \qquad M_y = \int_{\Omega_f} x p d\Omega_f. \tag{19.4}$$

To estimate the values P, M_x, M_y, we will use the reciprocity theorem. For this purpose we consider separately two punches with the same contact domain Ω_f penetrated without friction into an elastic half-space. The first standard punch has a plane bottom

$$z = f^0(x, y) = \alpha + \beta x + \gamma y, \quad (x, y) \in \Omega_f, \tag{19.5}$$

where α, β, and γ are some given constants. This punch is penetrated into an elastic half-space when the external loads at the domain Ω_q are absent, i.e.

$$q_x^0 = q_y^0 = q_z^0 = 0, \quad (x, y) \in \Omega_q. \tag{19.6}$$

The determined pressure p^0 in the contact domain ($(x, y) \in \Omega_f$) and the displacement components u^0, v^0, w^0 along the axes in the domain of the external force applications $(x, y) \in \Omega_q$ can be respectively represented as

$$p^0 = \alpha p_\alpha^0 + \beta p_\beta^0 + \gamma p_\gamma^0, \quad (x, y) \in \Omega_f \tag{19.7}$$

and

$$\begin{aligned} u^0 &= \alpha u_\alpha^0 + \beta u_\beta^0 + \gamma u_\gamma^0, \qquad (x, y) \in \Omega_q, \\ v^0 &= \alpha v_\alpha^0 + \beta v_\beta^0 + \gamma v_\gamma^0, \qquad (x, y) \in \Omega_q, \\ w^0 &= \alpha w_\alpha^0 + \beta w_\beta^0 + \gamma w_\gamma^0, \qquad (x, y) \in \Omega_q. \end{aligned} \tag{19.8}$$

The functions $p_\alpha^0(x, y)$, $p_\beta^0(x, y)$, $p_\gamma^0(x, y)$, $(x, y) \in \Omega_f$, and $u_\alpha^0(x, y)$, $u_\beta^0(x, y)$, …, $w_\gamma^0(x, y)$, $(x, y) \in \Omega_q$, do not depend on the constants α, β, γ.

The second punch with the desired shape $f = f(x, y)$, $(x, y) \in \Omega_f$, is penetrated into an elastic half-space in accordance with the boundary conditions (19.1)–(19.3) in the general case when

$$q_j \neq 0 \quad (j = x, y, z), \ (x, y) \in \Omega_q.$$

Thus, we have two systems of variables: the first system

$$\begin{cases} w^0 = f^0,\ p^0, & (x, y) \in \Omega_f, \\ u^0,\ v^0,\ w^0, & (x, y) \in \Omega_q, \end{cases}$$

that corresponds to the standard punch, and the second system

$$\begin{cases} w = f,\ p, & (x, y) \in \Omega_f, \\ q_x,\ q_y,\ q_z, & (x, y) \in \Omega_q, \end{cases}$$

that corresponds to the optimized punch. In accordance with the Betti reciprocity theorem [2, 3], we have

$$\int_{\Omega_f} f p^0 d\Omega_f = \int_{\Omega_f} f^0 p d\Omega_f + \int_{\Omega_q} \left(u^0 q_x + v^0 q_y + w^0 q_z\right) d\Omega_q. \tag{19.9}$$

Let us substitute the expressions (19.5)–(19.8) for f^0, p^0, u^0, v^0, w^0 into the relation (19.9) and perform elementary transformations taking into account the formulas (19.4). We obtain

$$P = \int_{\Omega_f} f p^0_\alpha d\Omega_f - \int_{\Omega_q} \left(u^0_\alpha q_x + v^0_\alpha q_y + w^0_\alpha q_z\right) d\Omega_q, \tag{19.10}$$

$$M_x = \int_{\Omega_f} f p^0_\gamma d\Omega_f - \int_{\Omega_q} \left(u^0_\gamma q_x + v^0_\gamma q_y + w^0_\gamma q_z\right) d\Omega_q, \tag{19.11}$$

$$M_y = \int_{\Omega_f} f p^0_\beta d\Omega_f - \int_{\Omega_q} \left(u^0_\beta q_x + v^0_\beta q_y + w^0_\beta q_z\right) d\Omega_q. \tag{19.12}$$

In what follows, we will use the following presentation for total force and moments applied to the optimized punch:

$$P = P_f(f) - P_q(q), \qquad M_j = M_j^f(f) - M_j^q(q), \quad j = x, y. \tag{19.13}$$

Here $P_f(f)$, $M_j^f(f)$ are the linear functionals of the desired punch shape

$$P_f(f) = \int_{\Omega_f} f p^0_\alpha d\Omega_f, \qquad M_x^f(f) = \int_{\Omega_f} f p^0_\gamma d\Omega_f, \qquad M_y^f(f) = \int_{\Omega_f} f p^0_\beta d\Omega_f \tag{19.14}$$

and the linear functionals $P_q(q)$, $M_j^q(q)$ depending on the loads q_x, q_y, q_z are expressed as

$$P_q(q) = \int_{\Omega_q} \left[u_\alpha^0 q_x + v_\alpha^0 q_y + w_\alpha^0 q_z\right] d\Omega_q,$$

$$M_x^q(q) = \int_{\Omega_q} \left[u_\gamma^0 q_x + v_\gamma^0 q_y + w_\gamma^0 q_z\right] d\Omega_q, \tag{19.15}$$

$$M_y^q(q) = \int_{\Omega_q} \left[u_\beta^0 q_x + v_\beta^0 q_y + w_\beta^0 q_z\right] d\Omega_q.$$

Using the presented expressions (19.10)–(19.15) for the total force and moments, we will consider the problems of the punch mass minimization under constraints on P, M_x, M_y. Thus, we will minimize the integral functional

$$J = \rho \int_{\Omega_f} \sqrt{1 + (\nabla f)^2}\, d\Omega_f = \rho S_f + \frac{\rho}{2} \int_{\Omega_f} (\nabla f)^2 d\Omega_f \longrightarrow \min_f \tag{19.16}$$

under the inequality constraints

$$P = P_f(f) - P_q(q) \geq P^*,$$

$$M_y = M_y^f(f) - M_y^q(q) \geq M_y^*,$$

$$M_x = M_x^f(f) - M_x^q(q) \geq M_x^*,$$

where the values P^*, M_y^*, M_x^* will be supposed as given.

19.3 Shape Optimization Based on the "Worst Case Scenario"

Suppose that all external loads q act in the domain Ω_q and their values are unknown beforehand, but their loads $q = \{q_x, q_y, q_z\}$ belong to some given set Λ_q, i.e.

$$q \in \Lambda_q.$$

In what follows, we will consider the following continuous set:

$$\Lambda_q = \left\{ q : q_j = q_j(x, y) \geq 0,\ j = x, y, z,\ (x, y) \in \Omega_q,\ \int_{\Omega_q} q_j(x, y) d\Omega_q \leq Q_j^* \right\}. \tag{19.17}$$

Here $Q_j^* \geq 0$ are specified positive constants. According to the inequalities in (19.17), any unidirectional forces, the resultants of which do not exceed specified values, can be applied to the region Ω_q.

Another method of describing the set Λ_q consists of a discrete specification of the set of possible forms of the forces, i.e.

$$\Lambda_q = \left\{ q = q^k(x, y) : k = 1, 2, \ldots, n,\ (x, y) \in \Omega_q \right\},$$

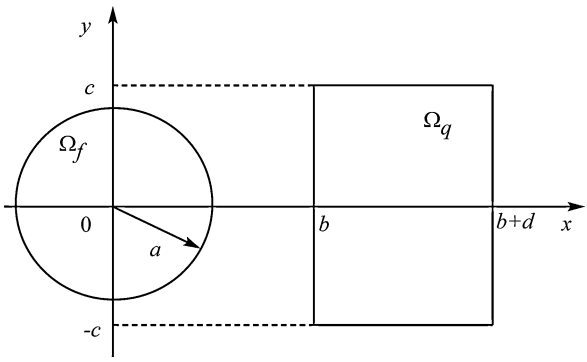


Fig. 19.1 The contact domain Ω_f and the region of application of the external loads Ω_q

where $q^k(x, y)$ are specified functions defining different cases of the loading of the region Ω_q. The "worst" force $q = q_{worst}$ is determined using the following inequality:

$$\max_{q \in \Lambda_q} \left(\max \left\{ P^* - P_f + P_q,\ M_x^* - M_x^f + M_x^q,\ M_y^* - M_y^f + M_y^q \right\} \right) \leq 0.$$

Let us consider the shape optimization problem where the constraints are imposed on the total force P and the moment M_y as

$$P \geq P^*, \qquad M_y - M_y^* \geq 0 \tag{19.18}$$

and assume that the contact domain Ω_f (the base of the optimized punch) has the shape of a circle

$$\Omega_f = \left\{ (x, y) : x^2 + y^2 \leq a^2 \right\} \tag{19.19}$$

and the region where the external load is applied has a rectangular shape (Fig. 19.1)

$$\Omega_q = \left\{ (x, y) : b \leq x \leq b + d,\ -c \leq y \leq c \right\}, \tag{19.20}$$

the point of which $(x = b,\ y = 0)$ nearest to Ω_f is a distance b from the origin of coordinates, i.e.

$$b = \min_{(x,y) \in \Omega_q} \sqrt{x^2 + y^2}, \quad \Omega_f \cap \Omega_q = 0. \tag{19.21}$$

Here a, b, c, and d are positive parameters, where $a < b$. The external forces considered are generated by distributed loads $q_z(x, y)$, which act along the z-axis, the resultant of which does not exceed the specified value Q_z^*, i.e.

$$q_z \in \Lambda_q = \left\{ q_z : q_z \geq 0, \quad \int_{\Omega_q} q_z d\Omega_q \leq Q_z^* \right\}. \tag{19.22}$$

In our analysis we will use the following expressions for $p_\alpha^0(x, y)$, $p_\beta^0(x, y)$, $(x, y) \in \Omega_f$, and $w_\alpha^0(x, y)$, $w_\beta^0(x, y)$, $(x, y) \in \Omega_q$, corresponding to the rigid punch

with the circular contact domain ($r = \sqrt{x^2 + y^2} \le a$) and a plane bottom:

$$\begin{aligned} p_\alpha^0 &= \frac{E}{\pi(1-\nu^2)\sqrt{a^2 - r^2}}, && (x, y) \in \Omega_f, \\ p_\beta^0 &= \frac{2Ex}{\pi(1-\nu^2)\sqrt{a^2 - r^2}}, && (x, y) \in \Omega_f, \\ w_\alpha^0 &= \frac{2}{\pi} \arcsin \frac{a}{r}, && (x, y) \in \Omega_q, \\ w_\beta^0 &= \frac{2x}{\pi}\left[\arcsin \frac{a}{r} - \frac{a}{r^2}\sqrt{r^2 - a^2}\right], && (x, y) \in \Omega_q. \end{aligned} \tag{19.23}$$

Here E and ν are respectively the Young modulus and the Poisson ratio of the elastic medium.

We will also use the expressions (19.19)–(19.23) and the following formulae:

$$q_{worst} = \arg\max_{q \in \Omega_q} \left(\max\left\{P^* - P_f + P_q, \quad M_y^* - M_y^f + M_y^q\right\}\right), \tag{19.24}$$

$$P_f = \int_{\Omega_f} f p_\alpha^0 d\Omega_f, \qquad P_q = \int_{\Omega_q} w_\alpha^0 q_z d\Omega_q,$$

$$M_y^f = \int_{\Omega_f} f p_\beta^0 d\Omega_f, \qquad M_y^q = \int_{\Omega_q} w_\beta^0 q_z d\Omega_q.$$

Using the guaranteed minimax approach (worst case scenario) employed in this section, the external load $q_z(x, y)$ is chosen from the admissible set (19.22) in calculating the worst case in accordance with (19.24). Taking also into account that the quantity w_α^0 is the decreasing function of the distance $r = \sqrt{x^2 + y^2}$ and the monotonic decrease in the function

$$w_\beta^0(r, \theta) = \frac{2}{\pi} r\left[\arcsin \frac{a}{r} - \frac{a}{r^2}\sqrt{r^2 - a^2}\right]\cos\theta, \quad r > a$$

as r increases for any θ in the interval $-\pi/2 < \theta < \pi/2$ (θ is the angle measured in the $x0y$ plane anticlockwise from the x-axis), and the representations (19.24), it can be shown that the "worst" load chosen from the admissible set of the loads (19.22) has the form of the Dirac δ-function (denoted by δ):

$$(q_z)_{worst} = Q_z^* \delta(x - b, y).$$

Thus, the "worst" load is a pointed force applied to the region Ω_q at the point ($x = b$, $y = 0$) nearest to the contact domain Ω_f.

The total load P and the moment M_y will be estimated as

$$P = \int_{\Omega_f} f(x,y)p_\alpha^0(x,y)d\Omega_f - \frac{2Q_z^*}{\pi}\arcsin\frac{a}{b} \geq P^*,$$
$$M_y = \int_{\Omega_f} f(x,y)p_\beta^0(x,y)d\Omega_f - \frac{2bQ_z^*}{\pi}\left(\arcsin\frac{a}{b} - \frac{a}{b^2}\sqrt{b^2-a^2}\right) \geq M_y^*, \tag{19.25}$$

when $q_z = (q_z)_{worst}$.

In the case where both constraints in (19.18) are "active", i.e. they are satisfied with the equality sign for $q_z = (q_z)_{worst}$, the finding of the optimal punch shape is reduced to the search for the extremum of the Lagrange augmented functional

$$J^L(f) = J(f) - \lambda\int_{\Omega_f} fp_\alpha^0 d\Omega_f - \mu\int_{\Omega_f} fp_\beta^0 d\Omega_f. \tag{19.26}$$

Here λ and μ are Lagrange multipliers corresponding to the constraints (19.25) taken with the rigorous equality sign.

The necessary condition for an extremum of the functional (19.26) and the corresponding boundary condition and the additional condition that the desired function f is limited as $r \to 0$ can be written in the form

$$\frac{\partial^2 f}{\partial r^2} + \frac{1}{r}\frac{\partial f}{\partial r} + \frac{1}{r^2}\frac{\partial^2 f}{\partial\theta^2} = -\frac{\lambda}{\rho}\varphi(r) - \frac{\mu}{\rho}\psi(r,\theta), \tag{19.27}$$
$$(f)_{r=a} = 0, \quad 0 \leq \theta \leq 2\pi, \tag{19.28}$$
$$\lim f < \infty, \quad r \to 0, \tag{19.29}$$

where θ is the polar angle measured in $x0y$ plane anticlockwise from the x-axis while the functions $\varphi(r)$ and $\psi(r,\theta)$ are given by the expressions

$$\varphi(r) = p_\alpha^0(r) = \frac{E}{\pi(1-\nu^2)\sqrt{a^2-r^2}},$$
$$\psi(r,\theta) = p_\beta^0(r,\theta) = \frac{2Er\cos\theta}{\pi(1-\nu^2)\sqrt{a^2-r^2}} = 2r\varphi(r)\cos\theta. \tag{19.30}$$

The Poisson equation (19.27) together with the representations (19.30) and the conditions (19.28), (19.29) constitute the boundary value problem for determining the desired shape of the punch. Its bounded solution can be written in the form

$$f(r,\theta) = \lambda\Phi(r) + \mu W(r,\theta),$$
$$\Phi(r) = \frac{E}{\rho\pi(1-\nu^2)}\left\{\sqrt{a^2-r^2} - a\ln\left(1+\sqrt{1-\frac{r}{a}}\right)\right\},$$
$$W(r,\theta) = \frac{2E\cos\theta}{3\rho\pi r(1-\nu^2)}\left\{a\left(a^2-r^2\right) - \left(a^2-r^2\right)^{3/2}\right\}.$$

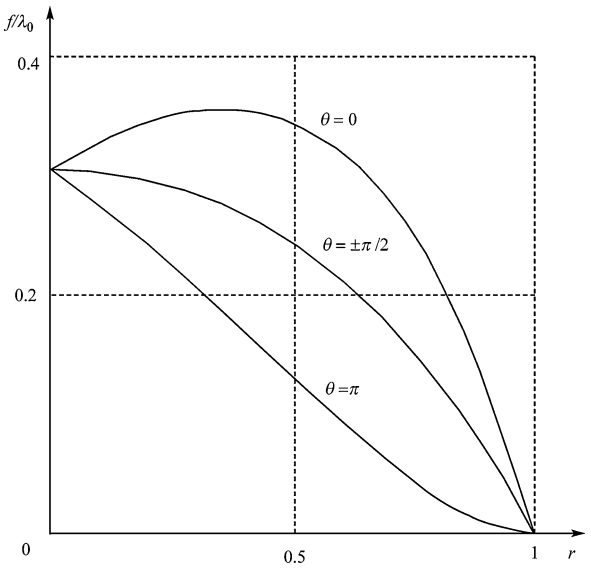


Fig. 19.2 The shape characteristic f/λ_0 against r for different values of θ

The Lagrange multipliers λ and μ are found from the system of two linear algebraic equations

$$\lambda \int_{\Omega_f} \varphi \Phi d\Omega_f + \mu \int_{\Omega_f} \varphi W d\Omega_f = P^* + \frac{2Q^*}{\pi} \arcsin \frac{a}{b},$$

$$\lambda \int_{\Omega_f} \psi \Phi d\Omega_f + \mu \int_{\Omega_f} \psi W d\Omega_f = M_y^* + \frac{2bQ^*}{\pi} \left(\arcsin \frac{a}{b} - \frac{a}{b^2} \sqrt{b^2 - a^2} \right).$$

In Fig. 19.2, we show graphs of f/λ_0 against r ($\lambda_0 = \lambda E/[\pi\rho(1-\nu^2)]$) for different values of θ for the case when $\mu = 0.8\lambda$. Note that the curve for $\theta = \pm\pi/2$ represents the shape of the punch in the case when the constraint on the moments is not imposed ($W(r,\theta) = 0$). In Fig. 19.3, we represent a corresponding 3-D shape of the optimal punch.

19.4 Shape Optimization Based on Probabilistic Formulation

In this section, the external forces $q = \{q_x, q_y, q_z\}$ applied to the domain Ω_q are considered as functions depending on random variables ξ and η, i.e.

$$q_j = q_j(x, y, \xi, \eta), \quad j = x, y, z, \ (x, y) \in \Omega_q. \tag{19.31}$$

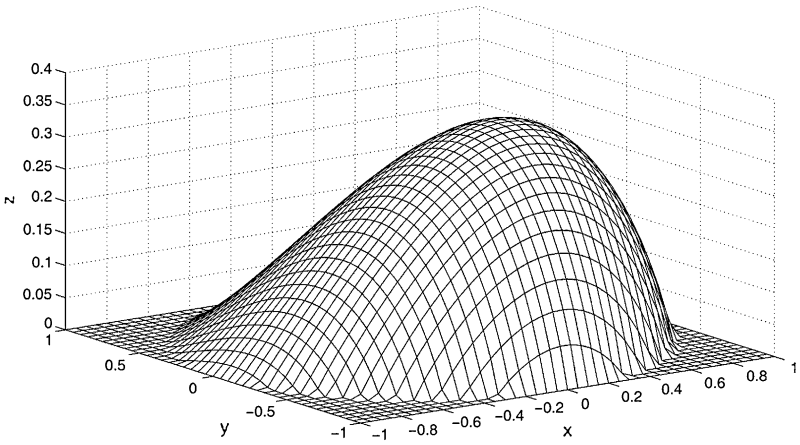


Fig. 19.3 The optimal punch shape

The random variables ξ, η are described by the joint probability density and the corresponding joint probability distribution functions $F(\xi, \eta)$, where

$$g(\xi, \eta) = \frac{\partial^2 F}{\partial \xi \partial \eta}.$$

Consider the case when the random external pointed loads

$$q_j(x, y, \xi, \eta) = Q_j \delta(x - \xi, y - \eta), \quad j = x, y, z$$

are applied to the domain

$$\Omega_q = \{x_1 \le x \le x_2,\ y_1 \le y \le y_2\}$$

and these loads are statistically independent, i.e.

$$g = g(\xi, \eta) = g_\xi(\xi) g_\eta(\eta)$$

and uniformly distributed, i.e. the multipliers $g_\xi(\xi)$, $g_\eta(\eta)$ of the joint probability density have the form

$$g_\xi(\xi) = \begin{cases} 0, & \xi < x_1, \\ 1/(x_2 - x_1), & x_1 < \xi < x_2, \\ 0, & \xi > x_2, \end{cases}$$

$$g_\eta(\eta) = \begin{cases} 0, & \eta < y_1, \\ 1/(y_2 - y_1), & y_1 < \eta < y_2, \\ 0, & \eta > y_2 \end{cases}$$

and the corresponding joint probability distribution functions $F_\xi(\xi)$, $F_\eta(\eta)$ are written as

$$F_\xi(\xi) = \begin{cases} 0, & \xi < x_1, \\ (\xi - x_1)/(x_2 - x_1), & x_1 < \xi < x_2, \\ 1, & \xi > x_2, \end{cases}$$

$$F_\eta(\eta) = \begin{cases} 0, & \eta < y_1, \\ (\eta - y_1)/(y_2 - y_1), & y_1 < \eta < y_2, \\ 1, & \eta > y_2, \end{cases}$$

where Q_j and x_1, x_2, y_1, y_2 are the given values ($x_1 < x_2$, $y_1 < y_2$).

Taking into account the boundary conditions (19.1)–(19.3) with the random forces (19.31), we can find random stresses σ_{xz}, σ_{yz}, σ_{zz} and random contact pressure distribution

$$p(x, y, \xi, \eta) = -\sigma_{zz}(x, y, \xi, \eta)$$

at the bottom of the punch ($(x, y) \in \Omega_f$) that can be used for determination of random resulting force P and moments M_x, M_y applied to the punch as

$$P(\xi, \eta) = \int_{\Omega_f} p(x, y, \xi, \eta) d\Omega_f,$$

$$M_x(\xi, \eta) = \int_{\Omega_f} y p(x, y, \xi, \eta) d\Omega_f,$$

$$M_y(\xi, \eta) = \int_{\Omega_f} x p(x, y, \xi, \eta) d\Omega_f.$$

Using the Betti reciprocity theorem [2, 3] and the corresponding relations (19.13)–(19.15), it is possible to estimate mathematical expressions of the total reaction force $\hat{P} = \mathscr{E}(P)$ and moments $\hat{M}_j = \mathscr{E}(M_j)$, $j = x, y$, evaluated with the help of the formulas

$$\begin{aligned} \hat{P} &= \mathscr{E}(P) = P_f(f) - \mathscr{E}\big(P_q(q)\big), \\ \hat{M}_j &= \mathscr{E}(M_j) = M_j^f(f) - \mathscr{E}\big(M_j^q(q)\big), \end{aligned} \tag{19.32}$$

where the mathematical expectation of the random function $e(\xi, \eta)$ is defined as

$$\mathscr{E}\big(e(\xi, \eta)\big) \equiv \int_{\Omega_q} g(\xi, \eta) e(\xi, \eta) d\xi d\eta.$$

Consider the probabilistic shape optimization problem for a rigid punch-shell consists of minimization of the mass functional (19.16) under the constraints

$$\hat{P} = P^*, \qquad \hat{M}_x = M_x^*, \qquad \hat{M}_y = M_y^* \tag{19.33}$$

imposed on the values of mathematical expectations of the total reaction force and moments. Here P^*, M_x^*, M_y^* are given problem parameters. In what follows in this section, we suppose that the contact domain Ω_f is circular with a given radius a, i.e.

$$\Omega_f = \left\{x^2 + y^2 \le a^2\right\} = \{0 \le r \le a,\ 0 \le \theta \le 2\pi\},$$

where r is the radius and θ is the angle measured in the xy plane from the x-direction.

To find the unknown punch shape, let us construct the Lagrange augmented functional corresponding to the optimization problem (19.16), (19.33) taking into account only the terms $P_f(f)$ and $M_j^f(f)$ in (19.32) that depend explicitly (in the form of the linear functional) on the shape f. We have

$$J^L = \int_{\Omega_f} \left[\frac{\rho}{2}(\nabla f)^2 - \lambda_\alpha p_\alpha^0 f - \lambda_\beta p_\beta^0 f - \lambda_\gamma p_\gamma^0 f\right] d\Omega_f, \tag{19.34}$$

where $r = \sqrt{x^2 + y^2}$,

$$\begin{aligned} p_\alpha^0(r) &= \frac{E}{\pi(1-\nu^2)\sqrt{a^2 - r^2}}, \\ p_\beta^0(r,\theta) &= \frac{2Er\cos\theta}{\pi(1-\nu^2)\sqrt{a^2 - r^2}}, \\ p_\gamma^0(r,\theta) &= \frac{2Er\sin\theta}{\pi(1-\nu^2)\sqrt{a^2 - r^2}}. \end{aligned} \tag{19.35}$$

To determine the Lagrange multipliers λ_α, λ_β, λ_γ in the augmented functional (19.34), we will use the conditions (19.33). The radius r and the angle θ belong to the domain $\Omega_f = \{0 \le r \le a,\ 0 \le \theta \le 2\pi\}$.

A necessary optimality condition and boundary condition for the desired function f constitute the following boundary value problem:

$$\begin{aligned} \Delta f \equiv \frac{\partial^2 f}{\partial r^2} + \frac{1}{r}\frac{\partial f}{\partial r} + \frac{1}{r^2}\frac{\partial^2 f}{\partial \theta^2} = \\ = -\frac{\lambda_\alpha}{\rho} p_\alpha^0(r) - \frac{\lambda_\beta}{\rho} p_\beta^0(r,\theta) - \frac{\lambda_\gamma}{\rho} p_\gamma^0(r,\theta), \end{aligned} \tag{19.36}$$

$$\left(f(r,\theta)\right)_{r=a} = 0, \quad 0 \le \theta \le 2\pi. \tag{19.37}$$

Using decomposition and shape functions, it is convenient to represent the desired bounded solution of the Poisson equation (19.36) with the Dirichlet condition (19.37) in the following form:

$$\begin{aligned} f(r,\theta) &= f_\alpha(r) + f_\beta(r,\theta) + f_\gamma(r,\theta), \\ f_\alpha(r) &= \lambda_\alpha \chi_\alpha(r), \end{aligned}$$

$$f_\beta(r,\theta) = \lambda_\beta \chi_\beta(r,\theta),$$
$$f_\gamma(r,\theta) = \lambda_\gamma \chi_\gamma(r,\theta).$$

Shape functions χ_i $(i = \alpha, \beta, \gamma)$ must satisfy the boundary value problems

$$\Delta \chi_i = -\frac{1}{\rho} p_i^0, \quad i = \alpha, \beta, \gamma, \tag{19.38}$$

$$(\chi_i)_{r=a} = 0, \qquad (\chi_i)_{r\to 0} < \infty \tag{19.39}$$

and Lagrange multipliers λ_i $(i = \alpha, \beta, \gamma)$ are determined with the help of the isoperimetric conditions (19.33). Taking into account that in the case $i = \alpha$ the function $p_\alpha^0 = p_\alpha^0(r)$ is axisymmetric with respect to the z-axis, we solve the boundary value problem (19.38), (19.39) and find a corresponding symmetric shape function

$$\chi_\alpha(r) = E\left[\sqrt{a^2 - r^2} - a \ln\left(\frac{a + \sqrt{a^2 - r^2}}{a}\right)\right] \Big/ \left(\rho\pi\left(1 - \nu^2\right)\right). \tag{19.40}$$

The optimal axisymmetric shape function (19.40) corresponds to the case where the only constraint on the external load is taken into account.

In the case where $i = \beta, \gamma$ the corresponding solutions of the boundary value problems (19.38) and (19.39) can also be found in analytical forms as

$$\chi_\beta(r,\theta) = 2E\cos\theta\left\{a\left(a^2 - r^2\right) - \left(a^2 - r^2\right)^{3/2}\right\}\left(3\pi\left(1 - \nu^2\right)\rho r\right),$$
$$\chi_\gamma(r,\theta) = 2E\sin\theta\left\{a\left(a^2 - r^2\right) - \left(a^2 - r^2\right)^{3/2}\right\}\left(3\pi\left(1 - \nu^2\right)\rho r\right).$$

The Lagrange multipliers λ_α, λ_β, λ_γ are found from the system of linear algebraic equations

$$\lambda_\alpha \delta_{\alpha\alpha} + \lambda_\beta \delta_{\beta\alpha} + \lambda_\gamma \delta_{\gamma\alpha} = P^* + C_P,$$
$$\lambda_\alpha \delta_{\alpha\gamma} + \lambda_\beta \delta_{\beta\gamma} + \lambda_\gamma \delta_{\gamma\gamma} = M_x^* + C_{M_x},$$
$$\lambda_\alpha \delta_{\alpha\beta} + \lambda_\beta \delta_{\beta\beta} + \lambda_\gamma \delta_{\gamma\beta} = M_y^* + C_{M_y}.$$

The coefficients δ_{ij} $(i = \alpha, \beta, \gamma;\ j = \alpha, \beta, \gamma)$ and the values C_P, C_{M_x}, C_{M_y} are defined as

$$\delta_{ij} = \int_{\Omega_f} \chi_i p_j^0 d\Omega_f,$$

$$C_P = \mathscr{E}\left\{\int_{\Omega_q} \left(u_\alpha^0 q_x + v_\alpha^0 q_y + w_\alpha^0 q_z\right) d\Omega_q\right\},$$

$$C_{M_x} = \mathscr{E}\left\{\int_{\Omega_q} \left(u_\gamma^0 q_x + v_\gamma^0 q_y + w_\gamma^0 q_z\right) d\Omega_q\right\},$$

$$C_{M_y} = \mathscr{E}\left\{\int_{\Omega_q} \left(u_\beta^0 q_x + v_\beta^0 q_y + w_\beta^0 q_z\right) d\Omega_q\right\},$$

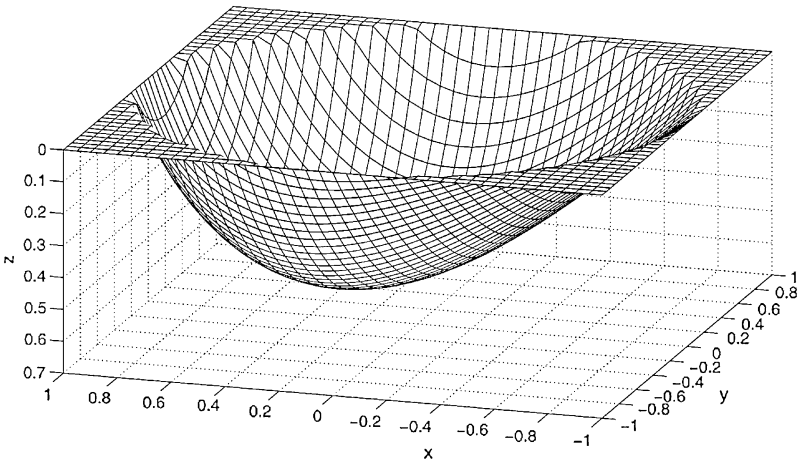


Fig. 19.4 The optimal punch shape

where p^{0_j}, $j = \alpha, \beta, \gamma$, w^0_α, w^0_β are given by the formulas (19.35), (19.23) and the functions u^0_α, u^0_β, u^0_γ, v^0_α, v^0_β, v^0_γ, w^0_γ are defined as

$$u^0_\alpha = -\frac{a(1-2\nu)x}{\pi(1-\nu)r^2},$$

$$u^0_\beta = -\frac{4a^3(1-2\nu)(x^2-y^2)}{3\pi(1-\nu)r^4},$$

$$u^0_\gamma = -\frac{8a^3(1-2\nu)xy}{3\pi(1-\nu)r^4},$$

$$v^0_\alpha = -\frac{a(1-2\nu)y}{\pi(1-\nu)r^2},$$

$$v^0_\beta = -\frac{8a^3(1-2\nu)xy}{3\pi(1-\nu)r^4},$$

$$v^0_\gamma = -\frac{4a^3(1-2\nu)(x^2-y^2)}{3\pi(1-\nu)r^4},$$

$$w^0_\gamma = \frac{2y}{\pi}\left(\arcsin\frac{a}{r} - \frac{a}{r^2}\sqrt{r^2-a^2}\right).$$

The optimal punch shape in the case of uniform distribution of probability density is presented in Fig. 19.4 when all three constraints are taken into account and $P^* = 1$, $M^*_x = M^*_y = 0.25$, $a = 1$.

19.5 Some Notes and Conclusions

Described in this investigation, the approaches to solving shape optimization problems in contact mechanics with uncertainties are based on the worst case scenario or on probabilistic formulation. Both the guaranteed approach and the probabilistic approach use the reciprocity relations for effective estimation of such integral characteristics as total forces and moments.

The procedure of the solution constructing consists of two parts. In the first part, the exact integral expressions for the total reactions are derived in the form of the linear functionals of the desired punch shape and the external loads given with incomplete data and successive finding of the "worst load" (the worst case scenario) or the mathematical expectations of the external actions (the probabilistic approach). In the second part, necessary optimality conditions and derived reciprocity relations are used for formulation of the boundary value problems for unknown shape functions. The solution of the boundary value problems and finding the desired punch shape are performed in an analytical manner.

Acknowledgements The work is performed under support of RFBR (grant 11-08-00030-a), RAS Program 13, Program of Support of Leading Scientific Schools (grant 3288.2010.1).

References

1. Banichuk NV (2009) Optimal forms in mechanics of contact interaction. Dokl Phys 54(7):333–337
2. Banichuk NV, Ivanova SYu (2009) Optimization problems of contact mechanics with uncertainties. Mech Based Des Struct Mach 37(2):143–156
3. Banichuk NV, Ivanova SYu (2009) Shape optimization in contact problems in the theory of elasticity with incomplete data on external actions. J Appl Math Mech 73(6):696–704
4. Banichuk NV, Ivanova SYu, Makeev EV, Ragnedda F (2009) Optimal stamp shape under probabilistic data concerning external loading. Probl Strength Plastic 71:52–60. In Russian
5. Banichuk NV, Neittaanmäki PJ (2010) Structural optimization with uncertainties. Solid mechanics and its applications, vol 162. Springer, Berlin
6. Banichuk NV, Ragnedda F, Serra M (2010) Some optimization problems for bodies in quasi-steady state wear. Mech Based Des Struct Mach 38(4):430–439
7. Haslinger J, Neittaanmäki P, Tiihonen T (1986) Shape optimization in contact problems based on penalization of the state inequality. Appl Math 31(1):54–77

Chapter 20
PPPC—Peer-2-Peer Streaming and Algorithm for Creating Spanning Trees for Peer-2-Peer Networks

Amir Averbuch, Yehuda Roditi, and Nezer Jacob Zaidenberg

Abstract We describe a system that builds peer-2-peer multicast trees. The proposed system has a unique algorithm that incorporates real-time and priority-based scheduler into a graph theory with robust implementation that supports multiple platforms. Special consideration was given to conditional access and trust computing. We also describe the system design as well as the computational aspects of processing the graphs used by the system.

20.1 Introduction

The bandwidth cost of live streaming prevents cost-effective broadcasting of rich multimedia content to Internet users.

For Internet streaming, the old-fashioned client-server model puts a considerable cost burden on the broadcaster. In the client-server model, a client sends a request to a server and the server sends a reply back to the client. This type of communication is at the heart of the IP [11] and TCP [12] protocol, and most of UDP [10] traffic as well. In fact, almost all upper layers of communication such as HTTP [5], FTP [1], SMTP [13], etc., implement the client-server models. The client-server communication model is known as unicast where a one-to-one connection exists between the client and the server. If ten clients ask for the same data at the same time, then ten exact replicas of the same replies will come from the server to each of the clients (as

A. Averbuch (✉)
School of Computer Science, Tel Aviv University, P.O. Box 39040, Tel Aviv 69978, Israel
e-mail: amir@math.tau.ac.il

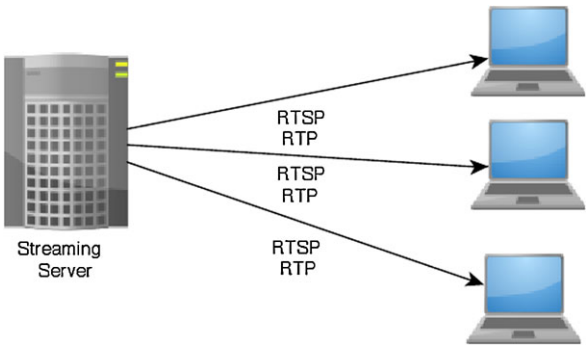
A. Averbuch · N.J. Zaidenberg
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland

N.J. Zaidenberg
e-mail: nezer.j.zaidenberg@jyu.fi

Y. Roditi
Academic College of Tel-Aviv-Yaffo, P.O. Box 8401, 61083 Tel Aviv, Israel
e-mail: jr@mta.ac.il

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_20,

Fig. 20.1 Unicast streaming



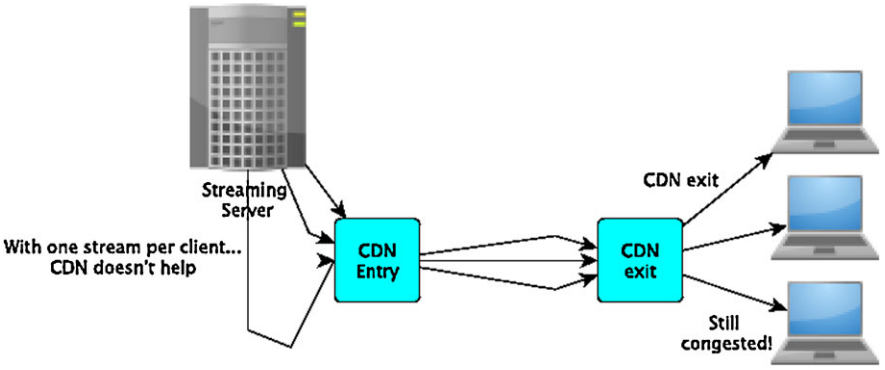


Fig. 20.2 CDN currently does not provide the solution

demonstrated in Fig. 20.1). This model remains the same regardless of the number of concurrent requests from the same number of unique clients, placing additional stress on the server with each additional user.

Furthermore, the problem exists to a much greater extent in live streaming scenarios with large crowds of listeners such as sport events, etc., as caching techniques such as proxies do not work with live streaming.

These problems also arise even when Content Delivery Networks (CDNs) are used for replicating static content to other servers at the edge of the Internet. Even when CDNs are used, every client is still served by one stream from a server, resulting in the consumption of a great amount of bandwidth (see Fig. 20.2). These infrastructure and cost challenges place a significant hurdle in front of existing and potential Web casters. While the media industry is seeking to bring streaming content with TV-like quality to the Internet, the bandwidth challenges often restrict a feasible, profitable business model.

In order to reduce the dependence on costly bandwidth, a new method of Internet communication called “multicast” was invented. Rather than using the one-to-one model of unicast, multicast is a “one-to-selected-many” communication method. However, multicast is not available on the current Internet infrastructure IPv4 and

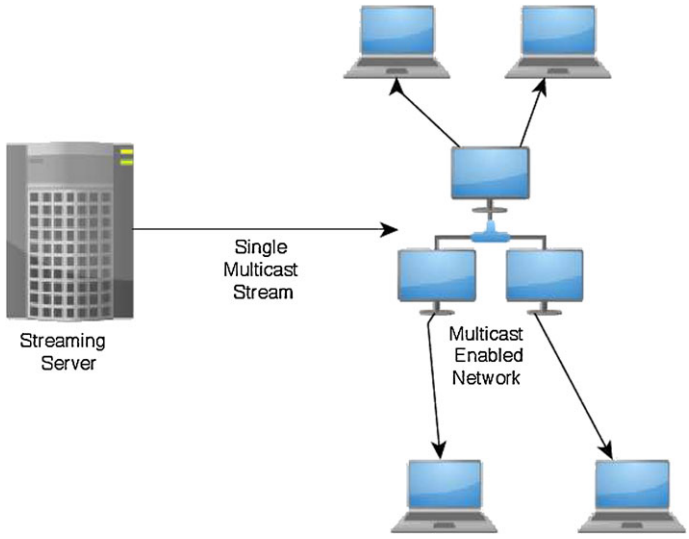


Fig. 20.3 Multicast streaming could provide a solution

may never be available outside private networks. An example of what multicast streaming looks like is demonstrated in Fig. 20.3.

A solution commonly proposed is to deploy Internet users as "broadcasters" using peer-2-peer [4, 7, 15] connections as a type of CDN.[1]

In this paper we describe our system for peer-2-peer streaming and our algorithm for handling network events.

20.2 System Design

The software industry has already anticipated the need for cost-effective, high-quality streaming and has developed applications that support multicast. Our peer-2-peer streaming system, called PPPC (Peer-2-Peer Packet Cascading) bypasses the lack of multicast in IPv4 Internet by providing multicast-like capabilities via peer-2-peer, and allows the use of the already available multicast software.

The concept of peer-2-peer streaming is a distributed architecture concept designed to use the resource of a client's (desktop computer) upstream in order to alleviate congestion in the broadcaster streaming server. (Using the client upstream does not affect its ability to surf or download files. The upstream resource is usually idle for most clients not involved in peer-2-peer activity (such as bittorrent [3]).)

In a peer-2-peer streaming system, the server only serves a fraction of selected simultaneous clients requesting the same stream and turns them into relay stations.

[1] Content delivery network.

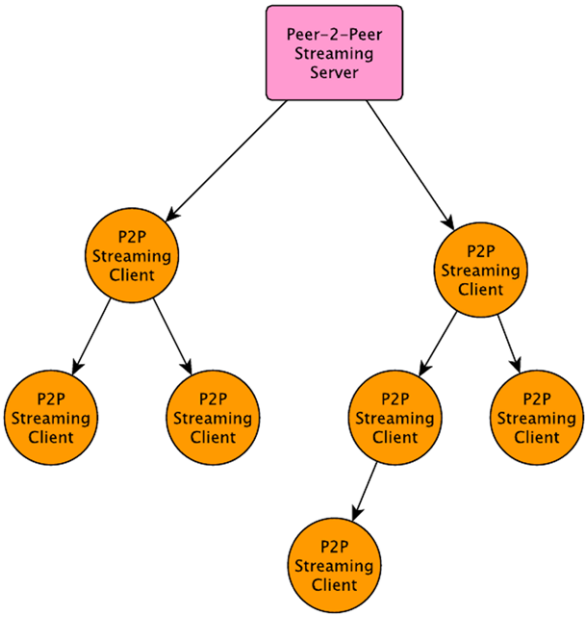


Fig. 20.4 Only peer-2-peer streaming solves the streaming problem on the Internet

Hereafter, the other clients who are requesting the same data will be served from one of the clients who received the stream first.

The clients shall only maintain a control connection to the server for receiving control input and reporting information. Also, we shall use every client as a sensor, to detect stream rate drops, to report the problem, and to complement the missing packet from either the PPPC router or another client. It is vital to detect any streaming issues in advance before the media player has started buffering or the viewer has noticed anything. Therefore, by following the peer-2-peer streaming concept and serving a fraction of the users, the server can serve a lot more users with the same bandwidth available. This is shown in Fig. 20.4.

Peer-2-peer packet cascading, or PPPC, is an implementation of the peer-2-peer concept to the streaming industry. PPPC provides a reliable multicasting protocol working on and above the standard IP layer. A PPPC system consists of the PPPC router and the PPPC protocol driver. The PPPC router stands between a generic media server, such as an MS Media server, a Real Media server or a QuickTime server, and the Viewers (see Fig. 20.5). The PPPC driver is in charge of the distribution of data and the coordination of the clients.

In a PPPC live stream, the PPPC router will receive a single stream from the media server and will route it directly to several "root clients". These clients will then forward the information to other clients and so on and so forth. Users connecting to each other will be relatively close network-wise. In this method, the data is cascaded down the tree to all the users while the PPPC router only serves directly (and pays bandwidth costs) for the root clients. As users that join and leave, the trees are dynamically updated. Moreover, the more users join the event, the more the PPPC

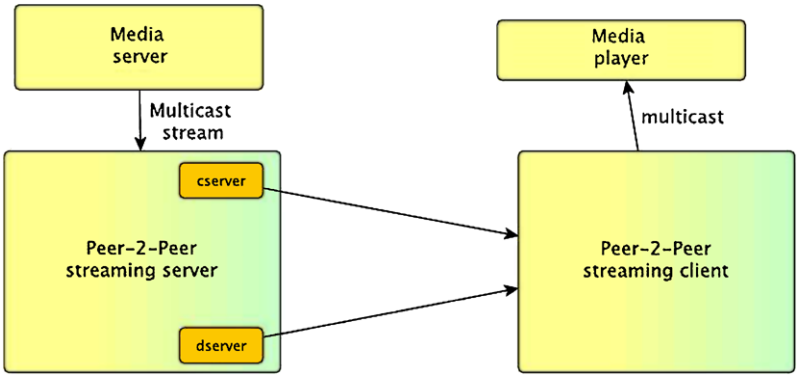


Fig. 20.5 PPPC data flow

router can build better trees saving even more, eliminating the financially undesirable linear connection between the cost of streaming and the number of users.

20.3 PPPC System Overview

Peer-2-peer packet cascading is a system designed to provide audio and video streaming clients with the capability to receive data from other clients and relay them to clients. The PPPC system is divided into a *PPPC router* and a *PPPC driver*. The PPPC router contains two logical components: the *Coordinating Server* (also called CServer) and the *Distributing Server* (also called DServer).

The PPPC driver installed on a client workstation (any standard PC) consists of thin client software that handles the reception and relay of the packets, and also "feeds" them to any media player. The client does not interact with a media player, it only delivers packets to the media player.

The coordinating server (CServer) is a command and control system in charge of all PPPC drivers listening to a single stream. The CServer is responsible for all the decisions in the system: For example, for a given client, from which client should it receive data and to which client should it transfer data, how should the tree be rebuilt after a new client arrives, what to do if a client in the middle of the tree is disconnected, and what happens when any given client reports on problems with receiving stream from his parent.

The distributing server (DServer) is a data replication and relay system. The DServer receives a multicast (data-only) stream and encapsulates the data in a PPPC format (recognized by PPPC driver). The DServer delivers the encapsulated packets to the roots of PPPC clients' trees (root clients). The CServer decides who the root clients are.

20.3.1 Data Flow in the PPPC System

In a PPPC system, a PPPC router must receive a data-only stream (i.e. no meta-data) from a generic media server and is responsible for delivering the stream to clients. In some way, a PPPC router acts very much like a multicast router. (A data-only stream is required because a stream with metadata will require the decoding of the stream and the right metadata to be sent to each of the clients thus missing the system's goal of generality.)

Most standard media servers can provide data-only stream, either directly or via a "multicast" option. The DServer in our system will receive the multicast stream or other data-only stream and pass it forward to the root clients. The PPPC drivers running on root clients' work stations pass the stream to other drivers on other clients. Therefore, each client acts as a small server, reusing the DServer code for this purpose.

When a PPPC driver, regardless of whether the PPPC driver also forwards the stream to other clients or not, receives the stream, it will forward it to the media player pretending to be the original media server using multicast or a fake IP if necessary. This is not real network traffic, but local traffic on the local host blocked in the kernel. Then, the media player will receive the stream and will act as if it received the stream directly from the media server. The PPPC driver will send a stream just like a media server.

Thus, the media server sends a standard (usually multicast) data stream, and the media player receives a standard stream. This enables the system to work with any media server, any media player and any codec, etc., without the need to have any specific integration.

20.3.2 Detailed Description of the System Components

One instance of the server handles one media stream. Multiple instances of the server are possible in order to handle more than one stream. Parts (entities) within the server communicate with each other by TCP enabling them to run on several computers.

20.3.2.1 Distributing Server (DServer)

The distributing server transfers the stream contents to root clients and serves as a backup source for clients (the DServer is also a backup server). It contains two physical components:

1. A single *receiver*, which gets the raw data from a media server via multicast or UDP. The DServer Receiver then encapsulates the data arriving from the media server in PPPC packets (recognized by PPPC clients).

2. One or more *distributors* which receive the information directly from the receiver and serve the root clients.

The distributors share packet relay and a connection code with the drivers, but they are the only entities that receive the stream directly from the receiver. This division is suggested in order to receive optimal scalability, and it allows the deployment of the distributors across several CDN sites.

20.3.2.2 Coordinating Server (CServer)

The coordinating server maintains the control connection with every client. It decides which clients connect between them, i.e., it constructs the PPPC tree. Our algorithm is implemented within the CServer. The CServer updates dynamically the PPPC tree on such events as connection/departure of clients, unexpected disconnections, etc.

The CServer, similar to the DServer, also contains two components:

1. A single centralized *main* module where all the users' (of a single stream) data is saved. The main module provides all the logic and decisions to the system.
2. One or more *proxies* who receive client connections and pass requests and responses to/from the CServer's main module.

In a large-scale PPPC system, where several proxies can exist, each maintains a connection to a large group of clients. The main module is the only place where complete information and decision making regarding all clients is kept for decision making regarding the clients' tree. Reports on the clients' connections and disconnections are handled by the main module.

20.3.2.3 The PPPC Driver

The PPPC driver is a very light client which consumes very little system resources apart from the relatively free upstream. It is the only client side software and it communicates with the CServer and DServer components and the PPPC drivers in other clients.

20.3.3 Viewing a Stream with PPPC—Life Cycle

This life cycle assumes that the clients select the stream using WWW:

1. The user accesses a page on the WWW which provides him with stream information.
2. The file is silently downloaded to the user's computer.

3. The PPPC driver parses the file which includes a standard media player activation file. The PPPC driver reads the CServer and DServer IP address as well as other parameters and invokes the standard media player to handle the stream.
4. The client connects simultaneously to the CServer and DServer.
5. The DServer sends data to the client which is immediately displayed to the user.
6. In a certain event[2] a CServer decides to rebuild the PPPC client trees.
7. The CServer sends the client messages with information about its new stream source (another client or the DServer) and possibly the address of other clients that should be served the stream.
8. A client connects to specified clients and starts to receive information from the parent and relays it to its children. The arrival of data is viewed through the media player to the user.
9. The CServer may decide during the operation to rebuild the tree and sends again the corresponding messages to the client, which disconnects its older connections and creates newer ones.
10. When the user decides to stop viewing the stream, the PPPC client recognizes it and sends the message "I'm going to disconnect" to the CServer and quits. Meanwhile, the CServer updates the clients' tree if necessary.

20.3.4 Maintaining a Certain Level of QoS

In order to maintain and guarantee a certain level of QoS, we will add a stream rate detection unit to every client. The stream rate is published by the CServer when clients join the stream. If a client detects that the stream rate has dropped below a certain level, he will be connected to the DServer to complement the missing packets or as an alternative stream source. Numerous reasons cause the packet rate to be dropped: parent disconnection (the packet rate drops to zero), a sudden drop in the packet rate when the parent uses his upstream to send an email, or a high CPU load on the parent machine. He might also report that his previous parent was a "bad parent"; then the CServer will not assign new children to a "bad parent".

The switch between parents and going to the DServer should be done very fast (within the streaming buffer time found in the client). If all packets arrive before the buffer expires, the user will never notice the switch between the parents.

We will describe the exact method in which bad parents are detected in Sect. 20.5.

20.4 Avoiding Flat Trees, Distress, and Guaranteeing a Certain Level of QoS

In this section we describe all the system engineering issues which are connected to the appearance of what we call "flat trees". Flat trees are trees that have a very

[2]For example, after some other new clients have arrived, or old clients have disconnected.

large number of root clients compared to the total number of clients and a very small number of peer-2-peer connections. We will also describe how these issues are solved.

We encountered several reasons for having extremely flat trees, and most of them were related to our goal to achieve a high level of QoS. This section describes our solution, which provides high streaming quality to clients who can receive the stream. This is done while we maintain a peer-2-peer connection with a high bandwidth saving ratio. We realized that QoS and the flat trees problem are closely connected. Several problems have been identified:

- Parents that could not serve clients constantly received new clients which caused a decrease in QoS.
- Parents that were declared bad parents never received new clients and caused flat trees.
- Clients that were unable to view the stream pass from parent to parent declared them all to be bad parents (hereby bad clients). Such a client can easily mark all clients in a tree as bad parents which will surely result in a flat tree.
- In case of a transmission problem in the upper layers of the tree, many clients in the lower layers of the tree did not receive the stream and reported their parents to be bad parents. This caused the multiplicity of bad parents.
- Clients that detected problems were generally not the direct children of the clients that caused the problem. Thus, many of the clients were declared to be bad for no apparent reason. (Same as above!)
- Due to the above conditions, the lower layers of the tree received poor-quality stream.

As we can see from above, most of the problems occurred due to faulty behavior when served by an unsatisfying packet rate. We shall hereby call this situation *distress*.

20.5 Bad Parents and Punishments

When a client reports to the CServer that his parent does not provide him with a sufficient packet rate, the CServer will mark the parent as a bad parent. In this case the bad parent's maximum number of children is set to its current child number.

The client that reported the bad parent will also connect to the DServer either to compliment the missing packets or to replace its current bad parent with the DServer. Therefore, a bad parent cannot have any more children. We will not disconnect any of the other children he already had. We will allow new children to replace one of the old ones if they were disconnected. If the previous client did not have any children then he will not have children anymore.

We "punished" bad parents so harshly to prevent any connection of new clients to them. Thus, we avoided a situation where a client connects to several "bad parents" before receiving the stream. Thus, the QoS is degraded. We provide another

punishment function that either produces no result, i.e. a client kept connecting to bad parents regardless of the punishment or the same result. The goal was that the client will never connect to bad parents.

The isolation of bad parents plays a very important role in guaranteeing a high QoS. We realized that a stream is never disrupted in real-world scenarios by the sudden disconnection of parents or fluctuations in their upstream. However, bad parents were often one of the main reasons for having flat trees. The clients could not find themselves a suitable parent because all possible parents were marked as bad parents and could not accept any new children.

Therefore, we give a chance for a bad parent to recover. We set a punishment time stamp where the punishment time is assigned to each of the bad parents. To recover from this situation we introduce bad parents' rehabilitation process (see Sect. 20.5.1). There are many temporary situations such as sending and e-mail which hogs the upstream, starting Microsoft Office, which causes a CPU surge for a couple of seconds, and many more. A "bad parent" can recover from the "temporary" situations. This should not prevent him from future productive service to clients.

20.5.1 Bad Parent Rehabilitation

There are many reasons for punishing a client and turning it into a bad parent. Nevertheless, we realized that the punishment mechanism on the PPPC network should be temporary. We shall associate a time stamp with the punishment time when a client is punished. After a period of time we will rehabilitate the parent and allow it to receive new connections.

The rehabilitation thread is in charge of bad parents rehabilitation. The suggested time period for rehabilitation is between 5 and 15 minutes.

20.5.2 Distress Logic: Marking of Bad Parents

A distress state is the state in which a client does not receive enough information within a PACKET_RATE_CALCULATION_PERIOD. There are two variables that dictate a distress state:

1. Parent distress is a boolean variable that indicates whether the parent sent any indication of entering into a distress state.
2. Current distress is a variable that may be equal to either no-distress, light distress, or major distress.

These variables introduce six different distress states:

No distress: The standard state. The packet rate is fine and the father has not informed otherwise.

Light distress: The state that occurs when a client receives less packets than DISTRESS_MIN_PACKET_RATE and there is no notification from the parent that he reached a similar state.
Parent distress: The parent indicates that he is in a light distress state but the information still flows fine.
Parent and light distress: Indicates that both the current client and its father experienced light distress.
Major distress: Indicates that the current packet rate is below MIN_PACKET_RATE.
Major and parent distress: Indicates that the current packet rate is below MIN_PACKET_RATE and the parent is also experiencing difficulties.

20.5.2.1 Entering into a Distress State

A packet rate threshold, DISTRESS_MIN_PACKET_RATE, is used to determine the upper bound of entering into a "light distress" state. A client in "light distress" does not complain about a bad parent, but opens a connection to the DServer to complement missing packets from there. The client only sends a "Bad Parent" message when the packet rate reaches MIN_PACKET_RATE, then it connects to the DServer (hereby major distress).

When a client enters into a distress state it will inform its children about its state. When a client enters into a major distress it will not report his parent as a bad parent if his parent is also in a distress state.

20.5.3 Bad Client

Some clients, for whatever reasons may be, are simply unable to receive the stream. Reasons may vary from insufficient downstream, congestion at the ISP or backbone, busy CPU, poor network devices or others.

Those clients will reach a major distress state regardless of the parent they were connected to. An "innocent" parent will be marked as a "bad" parent. In order to prevent this from happening we add new logic to the PPPC driver.

The client should stop complaining about bad parents when the problem is probably in its own ability to receive the stream.

20.6 The Algorithm

20.6.1 The Structure of the Internet—from Peer-2-Peer Streamer Perspective

Each of the Internet nodes viewing the stream comes from a location with an Internet connection. Often such organization is the user's home. The system we developed

is capable of detecting multiple nodes in the same location (such as two users in the same LAN or home) via multicast messages. The system ensure that at any location only one user will stream in or out of the location. This way we eliminate congestion and as a by-product guarantee that only one user in each location is visible to the algorithm.

Connections between Internet nodes tend to be lossy (it is typical to have about a 1 % packet loss) and add latency. Furthermore, not all connections are equal. When connecting to a "nearby" user, we can expect significantly less latency and packet loss then when connecting to a "far" user. Latency specifically is increased and can differ from a few milliseconds to hundreds of milliseconds depending on the distance.

We will now define *nearby* and *far* in Internet terms. Each Internet connection belongs to an "autonomous system". An autonomous system is usually an ISP[3] and sometimes a large company (such as HP or IBM) that is connected to at least two other autonomous systems. Two users from the same autonomous systems will be considered to be nearby each other.

We have created two additional levels of hierarchy. Below the autonomous system we have created a "subnet" level. Each IP address belongs to a subnet that defines a consistent range of IP addresses. Autonomous systems get their IP range as a disjoint union of subnets. Often each subnet belongs to different location that can be very far from each other (such as the east and west coast of the US). Thus, when possible, we prefer to connect to a user from the same subnet.

Autonomous systems are interconnected. Some autonomous systems can be considered "hubs" connected to many other autonomous systems. We have created "autonomous system families" centered on the hubs (containing all the autonomous systems that connect to the hub). When a user from the same autonomous system cannot be found, we will prefer a user from the same autonomous system family.

An autonomous system usually belongs to more than one autonomous system family. Thus, when choosing clients to connect to each other, we prefer clients that share a subnet. If none is found, we prefer clients that belong to the same autonomous system. If none is found, we prefer clients that belong to the same autonomous system family. Clients that have no shared container will be considered far and will not connect to each other.

20.6.2 Minimum Spanning Trees of Clients

The algorithm uses containers that hold all clients in a certain level. Since we can consider all clients that share a container and does not share any lower level container to be of an identical distance from each other, we can store all clients in a container in "heap-min" and only consider the best client in each heap to connect

[3] Internet service provider.

to. The best client will be considered using the distance from the source and the best available uplink (the best uplink considering other peers served by the client).

Algorithm 20.1 strives to maintain the graph as close to the MST as possible while responding to each new request (vertex joining, vertex removal) in nanoseconds. Indeed, our solution often involves finding an immediate solution such as connecting directly to the source and improves the solution over time until it reaches the optimal state. The proposed algorithm can handle very large broadcast trees (millions of listeners) in a nearly optimal state. As the server load increases (with more users), we may be further away from the optimal solution but we will not be too far and the stream quality for all users will be well tested.

Algorithm 20.1 Real time graph analysis

```
1: Read subnet to autonomous systems and autonomous systems to autonomous
   systems family files. Store information in a map.
2: Create global data structure spawn interface and parents rehabilitation thread
   and interface thread.
3: while Main thread is alive do
4:    if There are new requests then
5:       handle new request, touch at most 100 containers.
6:       Inform interface thread when you are done.
7:    else
8:       if there are dirty containers then
9:          clean at most 100 containers
10:         inform interface thread when you are done
11:      else
12:         wait for new request
13:      end if
14:   end if
15: end while
```

Clean and *dirty* in the algorithm sense are containers that are optimal and containers that are sub optimal for any reason.

For example, let us assume a client has disconnected. We will try to handle the disconnection by touching no more than 100 containers, let's say by connecting all the "child" nodes directly to the source. We will mark all the containers containing the nodes as dirty. At some point we will clean the container and fix any non-optimal state.

20.7 Related Systems

The authors have been involved with peer-2-peer streaming company vTrails Ltd that operated in peer-2-peer streaming scene in 1999–2002. (vTrails no longer operates.) Many of the concepts and system design may have originated from the authors' period with vTrails though the system has been written from scratch.

In recent years several peer-2-peer streaming systems have been proposed, some with a similar design. Our main contribution in this work is the peer-2-peer algorithm designed to calculate graph algorithms based on a real-time approach. Some features of the system approach such as the handling of distress state are also innovative.

ChunkySpeed [14] is a related system that also implements peer-2-peer multicast in a robust way. Unlike PPPC, ChunkySpeed does not take Internet distances into account.

ChunkCast [2] is another multicast over peer-2-peer system. ChunkCast deals with download time which is a different problem altogether. In streaming, a guaranteed constant bitrate is required. This requirement does not exist in content download which is not vulnerable to fluctuation in download speed and only the overall download time matters.

Climber [9] is a peer-2-peer stream based on an initiative for the users to share. It is our experience that user sharing is not the main problem but rather broadcasters willing to multicast their content on peer-2-peer networks. Thus Climber does not solve the real problem.

Microsoft [8] researched peer-2-peer streaming in a multicast environment and network congestion but had a completely different approach which involved multiple substreams for each stream based on the client abilities.

Liu et al. [6] recently researched peer-2-peer streaming servers' handling of bursts of crowds joining simultaneously which is handled by the algorithm easily thanks to its real time capabilities.

References

1. Bhushan AK (1971) File transfer protocol. RFC 114, April. Updated by RFC 133, RFC 141, RFC 171, RFC 172
2. Chun BG, Wu P, Weatherspoon H, Kubiatowicz J (2006) ChunkCast: an anycast service for large content distribution. In: Proceedings of the 5th international workshop on peer-to-peer systems (IPTPS)
3. Cohen B (2003) Incentives build robustness in BitTorrent. BitTorrent Inc., May. http://www2.sims.berkeley.edu/research/conferences/p2pecon/papers/s4-cohen.pdf
4. Feng C, Li B (2008) On large-scale peer-to-peer streaming systems with network coding. In: Proceedings of the 16th ACM international conference on multimedia, pp 269–278
5. Fielding R, Gettys J, Mogul J, Frystyk H, Masinter L, Leach P, Berners-Lee T (1999) Hypertext transfer protocol—HTTP/1.1. RFC 2616 (draft standard), June. Updated by RFC 2817, RFC 5785, RFC 6266
6. Liu F, Li B, Zhong L, Li B, Niu D (2009) How P2P streaming systems scale over time under a flash crowd? In: Proceedings of the 8th international workshop on peer-to-peer systems, IPSTS 09
7. Liu Y (2007) On the minimum delay peer-to-peer video streaming: how realtime can it be? In: Proceedings of the 15th international conference on multimedia, pp 127–136
8. Padmanabhan VN, Wang HJ, Chou PA (2005) Supporting heterogeneity and congestion control in peer-to-peer multicast streaming. In: Voelker GM, Shenker S (eds) Peer-to-peer systems III: third international workshop (IPTPS 2004). Lecture notes in computer science, vol 3279. Springer, Berlin, pp 54–63

9. Park K, Pack S, Kwon T (2008) Climber: an incentive-based resilient peer-to-peer system for live streaming services. In: Proceedings of the 7th international workshop on peer-to-peer systems, IPTPS 08
10. Postel J (1980) User datagram protocol. RFC 768 (Standard), August
11. Postel J (1981) Internet protocol. RFC 791 (Standard), September. Updated by RFC 1349
12. Postel J (1981) Transmission control protocol. RFC 793 (Standard), September. Updated by RFCs 1122, 3168, 6093
13. Postel J (1982) Simple mail transfer protocol. RFC 821 (Standard), August. Obsoleted by RFC 2821
14. Venkataraman V, Francis P, Calandrino J (2006) ChunkySpread: Multi-tree unstructured peer-to-peer multicast. In: Proceedings of the 5th international workshop on Peer-to-Peer systems, IPTPS 06
15. Zaidenberg NJ (2001) sFDPC—a P2P approach for streaming applications. MSc thesis, Tel Aviv University

Chapter 21
Safety Analysis and Optimization of Travelling Webs Subjected to Fracture and Instability

Nikolay Banichuk, Svetlana Ivanova, Matti Kurki, Tytti Saksa, Maria Tirronen, and Tero Tuovinen

Abstract The problems of safety analysis and optimization of a moving elastic web travelling between two rollers at a constant axial velocity are considered in this study. A model of a thin elastic plate subjected to bending and in-plane tension (distributed membrane forces) is used. Transverse buckling of the web and its brittle and fatigue fracture caused by fatigue crack growth under cyclic in-plane tension (loading) are studied. Safe ranges of velocities of an axially moving web are investigated analytically under the constraints of longevity and instability. The expressions for critical buckling velocity and the number of cycles before the fracture (longevity of the web) as a function of in-plane tension and other problem parameters are used for formulation and investigation of the following optimization problem. Finding the optimal in-plane tension to maximize the performance function of paper production is required. This problem is solved analytically and the obtained results are presented as formulae and numerical tables.

N. Banichuk (✉) · S. Ivanova
Ishlinsky Institute for Problems in Mechanics, Russian Academy of Sciences (RAS), Prospekt Vernadskogo 101, 119526 Moscow, Russia
e-mail: banichuk@ipmnet.ru

S. Ivanova
e-mail: ivanova@ipmnet.ru

M. Kurki · T. Saksa · M. Tirronen · T. Tuovinen
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland

M. Kurki
e-mail: matti.m.kurki@jyu.fi

T. Saksa
e-mail: tytti.saksa@jyu.fi

M. Tirronen
e-mail: maria.j.e.kuuluvainen@jyu.fi

T. Tuovinen
e-mail: tero.tuovinen@jyu.fi

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_21,

21.1 Introduction

Good runnability (performance) of webs and other axially moving bands and belts depends on the realized velocity and in-plane tension. Web breaks and instability are the most serious threats to good runnability. Arisen fracture and instability modes cause problems, e.g., for paper machines and printing presses. In practice, web instability in the form of buckling occurs when tension applied to the webs is less than some critical value, and extension of a safe stability range is realized by increasing the tension. However, a web break occurs when tension exceeds some critical value. Thus, the increase of the in-plane tension has opposite influences on the web stability and fracture. Both criteria are significant from the viewpoint of increased productivity demands, which mean faster production speeds and a longer safe production time interval (longevity).

Several studies related to the stable web movement exist in the literature. Vibrations of travelling membranes and thin plates were first studied by Archibald and Emslie [1], Miranker [11], Swope and Ames [18], Mote [12], Simpson [16], Chonan [4], and Wickert and Mote [22], concentrating on various aspects of free and forced vibrations. Stability of travelling rectangular membranes and plates was first studied by Ulsoy and Mote [19], Lin and Mote [9, 10], and Lin [8]. Recently, the behaviour of axially moving materials has been studied by, e.g., Shin et al. [15], Wang et al. [20], and Banichuk et al. [2].

In [2], buckling of an axially moving elastic plate was studied. The critical velocity and the corresponding buckling shapes were studied analytically as functions of problem parameters.

The field of fracture mechanics was developed by Irwin [7], based on the early papers of Inglis [6], Griffith [5], and Westergard [21]. Linear elastic fracture mechanics (LEFM), assuming a small plastic zone ahead of the crack tip, was first applied to paper material by Seth and Page [14], who measured fracture toughness of different paper materials. Swinehart and Broek [17] determined fracture toughness of paper using both the stress intensity factor and the strain energy release rate. They found that the measured crack length and fracture toughness were in a good agreement with the LEFM theory.

In this study, the product of critical buckling velocity and a safe time (longevity) will be taken as a maximized productivity function. We will evaluate analytically the performance criterion as a function of the applied tension and other problem parameters, and will study the problem of finding the optimal tension that maximizes the considered criterion.

21.2 Basic Relations and Formulation of the Optimization Problem

Consider an elastic web travelling at a constant velocity V_0 in the x direction and being simply supported by a system of rollers located at $x = 0, \ell, 2\ell, 3\ell, \ldots$

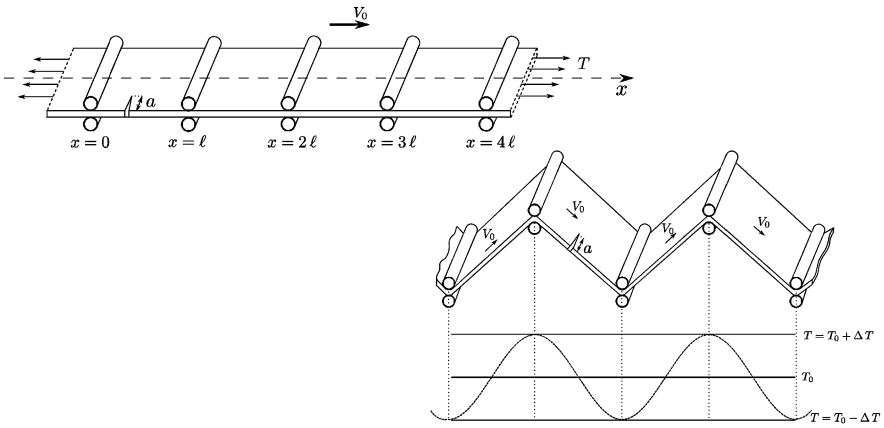


Fig. 21.1 *Top*: A travelling web having an initial crack, and being supported by a system of rollers. *Bottom*: A travelling web under cyclic tension, which is produced by the Earth's gravity

(Fig. 21.1). A rectangular element Ω_i, $i = 1, 2, 3, \ldots$, of the web

$$\Omega_i = \big\{(x, y) : i\ell \le x \le (i+1)\ell,\ -b \le y \le b\big\} \tag{21.1}$$

is considered in a Cartesian coordinate system, where ℓ and b are prescribed geometric parameters. Additionally, assume that the considered web is represented as an elastic plate having constant thickness h, the Poisson ratio ν, the Young modulus E, and bending rigidity D. The plate elements in (21.1) have small initial surface cracks (Fig. 21.1) of the length a with a given upper bound a_0, i.e.,

$$0 < a \le a_0,$$

and are subjected to homogeneous tension T, acting in the x direction.

The sides of the plate element ($i = 1, 2, 3, \ldots$)

$$\Gamma_\ell = \{x = 0,\ -b \le y \le b\} \quad \text{and} \quad \Gamma_r = \{x = \ell,\ -b \le y \le b\}$$

are simply supported, and the sides

$$\Gamma_- = \{y = -b,\ 0 \le x \le \ell\} \quad \text{and} \quad \Gamma_+ = \{y = b,\ 0 \le x \le \ell\}$$

are free of traction.

Consider the following scenario where the web is moving under cyclic in-plane tension and fatigue crack growth is realized. Suppose that the web is subjected to a cyclic tension T that varies in the given limits

$$T_{\min} \le T \le T_{\max},$$

where

$$T_{\min} = T_0 - \Delta T, \qquad T_{\max} = T_0 + \Delta T.$$

Above $\Delta T > 0$ is a given parameter such that

$$T_0 - \Delta T > 0 \quad \text{and} \quad \frac{\Delta T}{T_0} \ll 1. \tag{21.2}$$

For one cycle, the tension increases from $T = T_{\min}$ up to $T = T_{\max}$ (the loading process) and then decreases from $T = T_{\max}$ to $T = T_{\min}$ (the unloading process). The loading and unloading processes are supposed to be quasistatic: the dynamical effects are excluded.

The cyclic tension T may be produced by different imperfections. One cause of cyclic tension could be elastic vibrations of the rollers resulting in small changes in the distance between the rollers. In this case, the number of tension cycles may be very large. Another cause of cyclic tension could be the Earth's gravity [3] (see Fig. 21.1).

The product of the moving web velocity V_0 and the process time t_{f} can be considered a productivity criterion (performance function), i.e.,

$$J = m_0 V_0 t_{\mathrm{f}}, \quad m_0 = 2bm. \tag{21.3}$$

Here, m is the mass per unit area of the middle surface of the band. In (21.3), the velocity V_0 is taken from the safe interval

$$0 < V_0 < V_0^{\mathrm{cr}},$$

where V_0^{cr} is the critical buckling speed.

A safe interval for the safe functioning time (the number of cycles) is written as

$$0 < t_{\mathrm{f}} < t_{\mathrm{f}}^{\mathrm{cr}} \quad \text{or} \quad 0 < n < n^{\mathrm{cr}},$$

where $t_{\mathrm{f}}^{\mathrm{cr}}$ and n^{cr} are, respectively, the time interval and the total number of cycles before fatigue fracture. For a small cycle time period τ and a big number of cycles n, we assume that $t_{\mathrm{f}} = n\tau$ (approximately). Note that the critical buckling velocity V_0^{cr} and the critical functioning time $t_{\mathrm{f}}^{\mathrm{cr}}$ (the critical number of cycles n^{cr}) depend on the parameters of the average in-plane tension T_0, and the admissible variance ΔT, i.e.

$$V_0^{\mathrm{cr}} = V_0^{\mathrm{cr}}(T_0, \Delta T), \qquad t_{\mathrm{f}}^{\mathrm{cr}} = t_{\mathrm{f}}^{\mathrm{cr}}(T_0, \Delta T), \qquad n^{\mathrm{cr}} = n^{\mathrm{cr}}(T_0, \Delta T).$$

Consequently, the maximum value of the productivity criterion for the given values T_0 and ΔT is evaluated as

$$J(T_0, \Delta T) = m_0 V_0^{\mathrm{cr}}(T_0, \Delta T) t_{\mathrm{f}}^{\mathrm{cr}}(T_0, \Delta T) = m_0 \tau V_0^{\mathrm{cr}}(T_0, \Delta T) n^{\mathrm{cr}}(T_0, \Delta T).$$

The optimal average (mean) in-plane tension T_0 is found from a solution of the following optimization problem:

$$J^* = \max_{T_0} J(T_0, \Delta T). \tag{21.4}$$

To solve the formulated optimization problem (21.4), we will use the explicit analytical expressions for the values V_0^{cr} and n^{cr}. The value of T_0, giving the maximal production J^*, is denoted by T_0^*.

21.3 Evaluation of the Web Longevity and the Critical Buckling Velocity

To evaluate n^{cr}, let us apply the fatigue crack growth theory. Suppose that the web contains one initial crack of length a_0. The process of fatigue crack growth under cyclic tension (loading) can be described by the following equation [13] and the initial condition:

$$\frac{\mathrm{d}a}{\mathrm{d}n} = C(\Delta K)^k, \quad (a)_{n=0} = a_0. \tag{21.5}$$

Here the variance ΔK of the stress intensity factor K is determined with the help of formulae

$$\begin{gathered} \Delta K = K_{\max} - K_{\min}, \quad K_{\max} = \beta\sigma_{\max}\sqrt{\pi a}, \\ K_{\min} = \beta\sigma_{\min}\sqrt{\pi a}, \quad \sigma_{\max} = \frac{T_{\max}}{h}, \quad \sigma_{\min} = \frac{T_{\min}}{h}. \end{gathered} \tag{21.6}$$

In (21.5), C and k are material constants. In (21.6), h is the thickness of the web, n is the number of cycles, and $\sigma_{\max}$, $K_{\max}$, $\sigma_{\min}$ and $K_{\min}$ are, respectively, the maximum and minimum values of the stress σ and the stress intensity factor K in any given loading cycle. For the considered case, the surface crack geometric factor is $\beta = 1.12$.

Using (21.5) and (21.6), we write the crack growth equation in the following form:

$$\frac{\mathrm{d}a}{\mathrm{d}n} = C\kappa_0^k a^{k/2}, \quad \kappa_0 = \frac{2\beta\sqrt{\pi}}{h}\Delta T. \tag{21.7}$$

It follows from (21.7) and the initial condition in (21.5) that for considered values of the parameter $k \neq 2$, we will have

$$n = A\left[\frac{1}{a_0^{(k-2)/2}} - \frac{1}{a^{(k-2)/2}}\right], \quad A = \frac{2}{(k-2)C\kappa_0^k}. \tag{21.8}$$

Take into account that the unstable crack growth is obtained after $n = n^{\mathrm{cr}}$ cycles when the critical crack length a_{cr} satisfies the limiting relation

$$(K_{\max})_{a=a_{\mathrm{cr}}} = K_{\mathrm{C}},$$

or, in another form, we have

$$\beta\frac{T_{\max}}{h}\sqrt{\pi a_{\mathrm{cr}}} = K_{\mathrm{C}}. \tag{21.9}$$

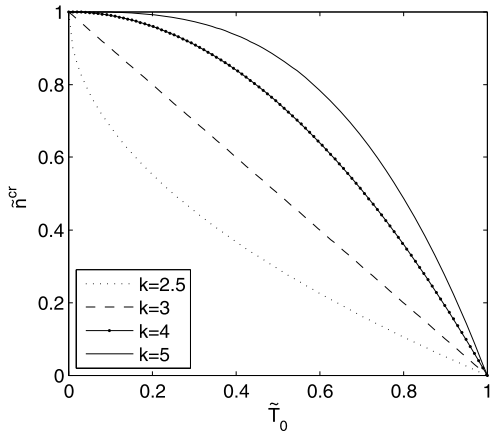


Fig. 21.2 Dependence of the (dimensionless) critical number of cycles $\tilde{n}^{\mathrm{cr}}$ on the (dimensionless) average tension $\tilde{T}_0$ for different values of the Paris constant k

Note that $\sigma_{\max}$ and $T_{\max}$ ($\sigma_{\min}$ and $T_{\min}$) are the maximum (minimum) stresses and tensions in the uncracked web, where the crack is located. Using (21.9) and the inequality $\Delta T/T_0 \ll 1$ in (21.2), we obtain

$$a_{\mathrm{cr}} = \frac{1}{\pi}\left(\frac{K_C h}{\beta T_{\max}}\right)^2 \approx \frac{1}{\pi}\left(\frac{K_C h}{\beta T_0}\right)^2$$

and, by (21.8), we will have the following expression for the critical number of cycles:

$$n^{\mathrm{cr}} = (n)_{a=a_{\mathrm{cr}}} = A\left[\frac{1}{a_0^{(k-2)/2}} - \left(\frac{\sqrt{\pi}\beta T_0}{K_C h}\right)^{k-2}\right]. \tag{21.10}$$

From the condition of positiveness of the expression in (21.10), we find the maximum value of admissible tensions:

$$T_0 \le \frac{1}{\sqrt{\pi a_0}}\frac{K_C h}{\beta} \equiv T_0^{\mathrm{M}}. \tag{21.11}$$

In the special case $k = 2$, we can find the critical number of cycles to be

$$n^{\mathrm{cr}} = B \ln\left[\frac{1}{\pi a_0}\left(\frac{K_C h}{\beta T_0}\right)^2\right], \quad B = \frac{1}{C\kappa_0^2}, \tag{21.12}$$

and the tension limit T_0^{M} is expressed by (21.11).

The dependence of the critical number of cycles n^{cr} on the average tension T_0 and the problem parameter k is shown in Fig. 21.2 using dimensionless quantities (defined below in (21.18) and (21.21)).

Stationary equations describing the behaviour of the web with the applied boundary conditions form the following eigenvalue problem (a buckling problem):

$$\begin{aligned}
\bigl(mV_0^2 - T_0\bigr)\frac{\partial^2 w}{\partial x^2} + D\biggl(\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4}\biggr) = 0, &\quad \text{in } \Omega,\\
w = 0, \qquad \frac{\partial^2 w}{\partial x^2} = 0, &\quad \text{on } \Gamma_\ell \text{ and } \Gamma_r,\\
\frac{\partial^2 w}{\partial y^2} + \nu\frac{\partial^2 w}{\partial x^2} = 0, &\quad \text{on } \Gamma_- \text{ and } \Gamma_+,\\
\frac{\partial^3 w}{\partial y^3} + (2-\nu)\frac{\partial^3 w}{\partial x^2 \partial y} = 0, &\quad \text{on } \Gamma_- \text{ and } \Gamma_+.
\end{aligned} \tag{21.13}$$

Here $D = Eh^3/(12(1-\nu^2))$, and m is the mass per unit area of the middle surface of the plate, and we denote the eigenvalue

$$\lambda = \gamma^2 = \frac{\ell^2}{\pi^2 D}\bigl(mV_0^2 - T_0\bigr).$$

The critical instability (buckling mode) velocity of the travelling plate, as was shown by [2], is given by

$$\bigl(V_0^{\mathrm{cr}}\bigr)^2 = \frac{T_0}{m} + \frac{\gamma_*^2}{m}\frac{\pi^2 D}{\ell^2}, \tag{21.14}$$

where $\gamma_*^2 = \lambda_*$ is the minimal eigenvalue of the problem (21.13). The parameter $\gamma = \gamma_*$ is found as the root of the equation (see Fig. 21.3)

$$\Phi(\gamma, \mu) - \Psi(\gamma, \nu) = 0, \tag{21.15}$$

where

$$\begin{aligned}
\Phi(\gamma, \mu) &= \tanh\biggl(\frac{\sqrt{1-\gamma}}{\mu}\biggr)\coth\biggl(\frac{\sqrt{1+\gamma}}{\mu}\biggr),\\
\Psi(\gamma, \nu) &= \frac{\sqrt{1+\gamma}}{\sqrt{1-\gamma}}\frac{(\gamma+\nu-1)^2}{(\gamma-\nu+1)^2}, \quad \mu = \frac{\ell}{\pi b}.
\end{aligned} \tag{21.16}$$

As it is seen from (21.15) and (21.16), the root $\gamma = \gamma_*$ depends on ν and μ and does not depend on the other problem parameters, including the value of tension T_0. Consequently, the critical instability velocity, defined in (21.14), is increased with the increasing of tension T_0. However, the increasing of T_0 is limited due to initial damages and other imperfections.

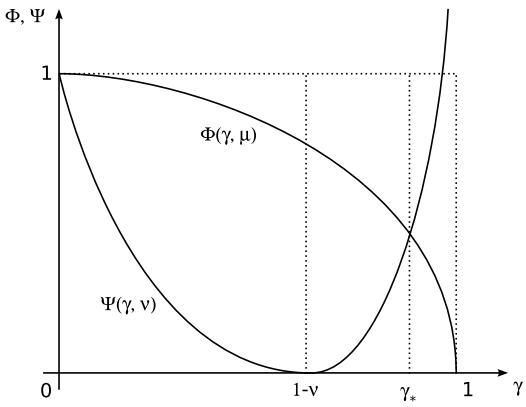


Fig. 21.3 Behaviour of Φ and Ψ as functions of γ

21.4 Optimization and Performance Function

The most important factor for runnability and stability of moving bands, containing initial imperfections, is the applied tension. To find a safe and optimal T_0 maximizing the performance function is our considered problem. To perform this task, let us represent the optimized functional (21.3) as a function of the average tension T_0. If we take into account explicit expressions for n^{cr}, in (21.10), and for V_0^{cr}, in (21.14), use (21.3), and perform necessary algebraic transformations, assuming that $k \neq 2$, we will have

$$J(T_0) = m_0 \tau V_0^{\mathrm{cr}}(T_0) n^{\mathrm{cr}}(T_0) = J_0 \left[1 + \frac{1}{D}\left(\frac{\ell}{\gamma_* \pi}\right)^2 T_0\right]^{1/2} \left[1 - \left(\frac{\beta\sqrt{\pi a_0}}{hK_{\mathrm{C}}} T_0\right)^{k-2}\right],$$

where

$$J_0 = \frac{4b\tau\pi a_0 \gamma_* \sqrt{Dm}}{(k-2)C\ell} \left(\frac{h}{2\beta \Delta T \sqrt{\pi a_0}}\right)^k. \tag{21.17}$$

The performance function J is proportional to the multiplier J_0 and, consequently, the optimized tension T_0 does not depend on this parameter.

For convenience of the following estimations and reduction of characteristic parameters, we introduce the dimensionless values

$$\tilde{J} = \frac{J}{J_0}, \qquad \tilde{T}_0 = \frac{T_0}{T_0^{\mathrm{M}}} = \frac{\beta\sqrt{\pi a_0}}{K_{\mathrm{C}} h} T_0, \qquad g = \frac{K_{\mathrm{C}} h}{\beta D \sqrt{\pi a_0}} \left(\frac{\ell}{\gamma_* \pi}\right)^2, \tag{21.18}$$

and represent the optimized functional and the interval of optimization as

$$\tilde{J}(\tilde{T}_0) = (1 + g\tilde{T}_0)^{1/2} \left(1 - \tilde{T}_0^{k-2}\right), \quad k > 2 \tag{21.19}$$

with

$$0 < \tilde{T}_0 < 1. \tag{21.20}$$

In other words,

$$\tilde{J}(\tilde{T}_0) = \tilde{V}_0^{\mathrm{cr}}(\tilde{T}_0)\,\tilde{n}^{\mathrm{cr}}(\tilde{T}_0)$$

with

$$\tilde{V}_0^{\mathrm{cr}}(\tilde{T}_0) = (1 + g\tilde{T}_0)^{1/2} \quad \text{and} \quad \tilde{n}^{\mathrm{cr}}(\tilde{T}_0) = 1 - \tilde{T}_0^{k-2}. \tag{21.21}$$

In the special case $k = 2$, we will use the expressions (21.3), (21.12) and (21.14) and perform algebraic transformations. We will have

$$J(T_0) = m_0 \tau V_0^{\mathrm{cr}}(T_0) n^{\mathrm{cr}}(T_0) = J_1 \left[1 + \frac{1}{D}\left(\frac{\ell}{\gamma_* \pi}\right)^2 T_0\right]^{1/2} \ln\left(\frac{hK_{\mathrm{C}}}{\beta\sqrt{\pi a_0}} \frac{1}{T_0}\right)$$

with

$$J_1 = \frac{4b\tau\pi\gamma_*\sqrt{Dm}}{C\ell}\left(\frac{h}{2\beta\Delta T\sqrt{\pi}}\right)^2.$$

Using the dimensionless values $\tilde{J} = J/J_1$ and $\tilde{T}_0$, g from (21.18), we find

$$\tilde{J}(\tilde{T}_0) = \ln\left(\frac{1}{\tilde{T}_0}\right)(1 + g\tilde{T}_0)^{1/2}, \quad 0 < \tilde{T}_0 < 1. \tag{21.22}$$

It is seen from (21.22) that

$$0 = (\tilde{J})_{\tilde{T}_0=1} \le \tilde{J}(\tilde{T}_0) \le (\tilde{J})_{\tilde{T}_0=0} = \infty, \quad 0 < \tilde{T}_0 < 1. \tag{21.23}$$

Note that (21.23) also holds in the case $k < 2$, when

$$\tilde{J}(\tilde{T}_0) = -(1 + g\tilde{T}_0)^{1/2}\left(1 - \tilde{T}_0^{k-2}\right)$$

and

$$J_0 = \frac{4b\tau\pi a_0\gamma_*\sqrt{Dm}}{(2-k)C\ell}\left(\frac{h}{2\beta\Delta T\sqrt{\pi a_0}}\right)^k.$$

Thus, in the case $k \le 2$, the optimum is $\tilde{T}_0 = 0$, meaning that the model omits the effect of the critical speed. However, for most materials $k \approx 3$ or bigger.

21.5 Results and Discussion

The optimization problem (21.19)–(21.20) was solved numerically for different values of k: for $k = 2.5$, $k = 3$, and $k = 3.5$. The material parameters were chosen to describe a paper material. Young's modulus was $E = 10^9$ Pa, the Poisson ratio was $\nu = 0.3$, the mass per unit area was $m = 0.08$ kg/m^2, and the strain energy rate over density was $G_{\mathrm{C}}/\rho = 10$ J m/kg. The size of the rectangular element (Ω_i) was $\ell \times 2b = 0.1$ m $\times$ 10 m, and the surface crack geometric factor was $\beta = 1.12$. The

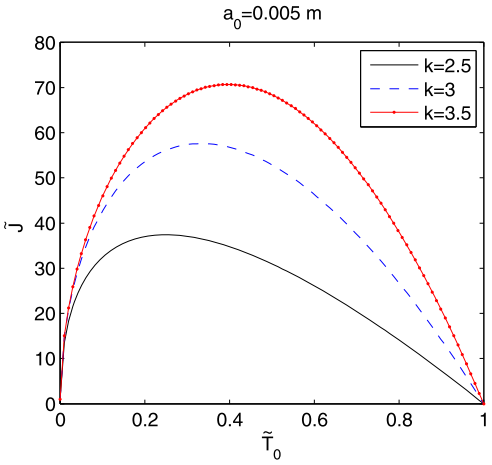


Fig. 21.4 Performance ($\tilde{J}$) dependence on tension ($\tilde{T}_0$) (dimensionless quantities)

Table 21.1 Dependence of the optimum of $\tilde{J}$ (performance) on the parameters k (Paris constant) and a_0 (m, initial crack length)

$\tilde{J}*$		a_0 (m)			
		0.005	0.01	0.05	0.1
k	2.5	37.4023	31.4527	21.0369	17.6920
	3	57.5834	48.4230	32.3862	27.2358
	3.5	70.6836	59.4390	39.7532	33.4308
		g			
		2.2379×10^4	1.5824×10^4	7.0768×10^3	5.0201×10^3

material constants in (21.5) were $k = 2.5,\ 3,\ 3.5$, and $C = 10^{-14}$. Paper fracture toughness K_C was calculated from the relation $G_C = K_C^2/E$ [7]. The variance in tension was chosen to be small, $\Delta T = 0.1$ N/m. The investigated values of initial crack lengths were $a_0 = 0.005$ m, 0.01 m, 0.05 m, 0.1 m. As illustrated in Fig. 21.1, the length of one cycle was assumed to be 2ℓ. This value was used to approximate the cycle time period τ by $\tau = 2\ell/V_0^{cr}$ after the value of V_0^{cr} was evaluated by the optimization.

In Fig. 21.4, the dimensionless performance function (21.19) is plotted for $k = 2.5,\ 3,\ 3.5$. It is seen that the value of optimal tension $\tilde{T}_0^*$ is increased with increasing the value of k.

In Tables 21.1 and 21.2, the results of the non-dimensional optimization problem (21.19)–(21.20) are shown for the considered values of the parameters k and a_0. In Table 21.1, the values of the productivity function $\tilde{J}$ at the optimum are shown. It can be noted that an increase in the length of the initial crack a_0 decreases productivity. The values of productivity seem to increase when k is increased. However,

Table 21.2 Dependence of the optimal tension $\tilde{T}_0^*$ on the parameters k (Paris constant) and a_0 (m, initial crack length)

$\tilde{T}_0^*$		a_0 (m)			
		0.005	0.01	0.05	0.1
k	2.5	0.2500	0.2499	0.2499	0.2498
	3	0.3333	0.3333	0.3332	0.3332
	3.5	0.3968	0.3968	0.3968	0.3967

Table 21.3 *Left*: Dependence of the optimal tension T_0^* (N/m) on the parameters k (Paris constant) and a_0 (m, initial crack length). *Right*: Critical velocity V_0^{cr} (m/s) at the optimum, depending on the parameters k and a_0

T_0^* (N/m)		a_0 (m)			
		0.005	0.01	0.05	0.1
k	2.5	504	356	159	113
	3	672	475	212	150
	3.5	800	565	253	179

$V_0^{\mathrm{cr}}(T_0^*)$ (m/s)		a_0 (m)			
		0.005	0.01	0.05	0.1
k	2.5	79.352	66.727	44.623	37.523
	3	91.628	77.051	51.529	43.332
	3.5	99.979	84.073	56.226	47.282

Table 21.4 *Left*: Dependence of the optimum of J (kg, performance) on the parameters k (Paris constant) and a_0 (m, initial crack length). *Right*: The number of cycles n^{cr} at the optimum, depending on the parameters k and a_0

J^* (kg)		a_0 (m)			
		0.005	0.01	0.05	0.1
k	2.5	121168	101894	68151	57315
	3	4821	3409	1525	1078
	3.5	216	128	38	23

$n^{\mathrm{cr}}(T_0^*)$		a_0 (m)			
		0.005	0.01	0.05	0.1
k	2.5	757300	636834	425943	358216
	3	30130	21306	9529	6738
	3.5	1348	801	239	142

one must take into account that also J_0, in (21.17), depends on k, which affects the actual productivity $J = J_0\tilde{J}$. In Table 21.2, the optimal values of the dimensionless tension $\tilde{T}_0^*$ are shown. It is seen that the dimensionless tension values slightly decrease when the crack size is increased.

Since the actual optimal productivity, the actual tension, and the related critical speed and the critical number of cycles are of interest, these values were found at the optimum and are shown in Tables 21.3 and 21.4. Note that several assumptions have been made. Firstly, the Paris constant $C = 10^{-14}$ is assumed to be independent of k, and both of the values are not measured for paper but were chosen to be close to the typical values of some known materials. Secondly, the cycle time period τ is approximated assuming that one cycle length is 2ℓ, and using the relation, $\tau = 2\ell/V_0^{\mathrm{cr}}$.

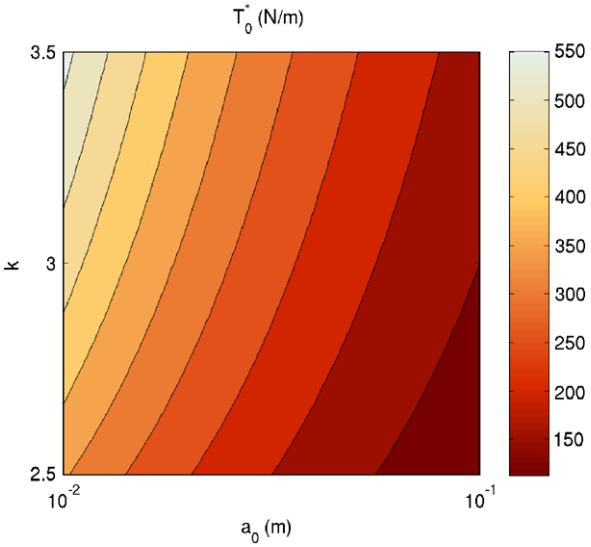


Fig. 21.5 A colorsheet showing the dependence of the optimal tension T_0^* (N/m) on the parameters k (Paris constant) and a_0 (m, initial crack length). Note the logarithmic scale of a_0

The actual optimal tension T_0^* is calculated from (21.18), that is $T_0^* = T_0^{\mathrm{M}} \tilde{T}_0^*$. Since T_0^{M} only depends on fixed values, and the material parameters in T_0^{M} are measured and known for paper material, the results for the actual optimal tension, shown in Table 21.3 (left), are comparable and quite reliable. The results for the optimal tension T_0^* are also illustrated as a colorsheet in Fig. 21.5.

In Table 21.3 (right), the critical velocities corresponding to the optimal values of tension $V_0^{\mathrm{cr}}(T_0^*)$ are shown. The values of velocities can be calculated directly from (21.14) using the values in Table 21.3 (left). As expected, the velocities decrease as a_0 is increased.

The actual optimal number of cycles $n^{\mathrm{cr}}(T_0^*)$ and the actual optimal productivity J^* are more difficult to predict, since they depend on the Paris constant C, which is not known for paper materials. As mentioned above, the same value of C, $C = 10^{-14}$, is used for all investigated values of k, which may not be reasonable. Since the value of κ_0 defined in (21.7) is big (in this case $\Delta T > h$), then κ_0^k increases with the increase in k. Keeping C constant, we see from (21.7) that the crack growth rate may be bigger with a bigger value of k depending on the value of $a^{k/2}$, which is small. This means that the number of cycles may be the smaller the greater the value of k is, which can also be seen from (21.8): the greater the value of k, the smaller the value of A. In the results in Table 21.4 (right), it can be seen that the effect of κ_0 is big, and the number of cycles at the optimum decreases remarkably when k is increased. This also results in a decrease in the optimal productivity J^*, which is shown in Table 21.4 (left).

Comparing the results in Tables 21.1 and 21.4 (left), we therefore make no conclusion about the effect of k on the actual performance J^*. The qualitative result of the decrease in the performance J^* when a_0 is increased is, however, reported.

21.6 Conclusion

In this study, the problems of safety analysis and optimization of a moving elastic web travelling between two rollers at a constant axial velocity were investigated. Instability of the web (transverse buckling) and its fatigue crack growth under a cyclic in-plane tension were included in the study. The expressions for the critical buckling velocity and the number of cycles before the fracture (longevity of the web) as a function of in-plane tension and other problem parameters were used to formulate analytically an optimization problem, in which the productivity was maximized. The optimal tension maximizing the productivity function was found.

The optimal values of tension seemed to be very sensitive to the length of the initial crack. It was found that the greater the initial crack, the smaller the optimal tension and, consequently, the smaller the maximal productivity.

It should be noted that the critical velocity of the (paper) web was considered in vacuum, and the effects of the surrounding fluid were excluded in this study, and remain as topics for future research. Thus, the results are to be interpreted as approximate.

Acknowledgements This research was supported by the Academy of Finland (grant no. 140221) and the Jenny and Antti Wihuri Foundation.

References

1. Archibald FR, Emslie AG (1958) The vibration of a string having a uniform motion along its length. J Appl Mech 25:347–348
2. Banichuk N, Jeronen J, Neittaanmäki P, Tuovinen T (2010) On the instability of an axially moving elastic plate. Int J Solids Struct 47(1):91–99
3. Banichuk N, Jeronen J, Saksa T, Tuovinen T (2011) Static instability analysis of an elastic band travelling in the gravitational field. J Struct Mech 44(3):172–185
4. Chonan S (1986) Steady state response of an axially moving strip subjected to a stationary lateral load. J Sound Vib 107(1):155–165
5. Griffith AA (1921) The phenomena of rupture and flow in solids. Philos Trans R Soc Lond Ser A, Math Phys Sci 221:163–198
6. Inglis CE (1913) Stresses in a plate due to the presence of cracks and sharp corners. Trans Instit Naval Architect 55:219–241
7. Irwin GR (1958) Fracture. In: Flügge S (ed) Handbuch der Physik, vol VI. Springer, Berlin, pp 551–590
8. Lin CC (1997) Stability and vibration characteristics of axially moving plates. Int J Solids Struct 34(24):3179–3190
9. Lin CC, Mote CD (1995) Equilibrium displacement and stress distribution in a two-dimensional, axially moving web under transverse loading. J Appl Mech 62(3):772–779
10. Lin CC, Mote CD (1996) Eigenvalue solutions predicting the wrinkling of rectangular webs under non-linearly distributed edge loading. J Sound Vib 197(2):179–189
11. Miranker WL (1960) The wave equation in a medium in motion. IBM J Res Dev 4(1):36–42
12. Mote CD (1972) Dynamic stability of axially moving materials. Shock Vib Dig 4(4):2–11
13. Paris PC, Erdogan F (1963) A critical analysis of crack propagation laws. J Basic Eng 85(4):528–534
14. Seth RS, Page DH (1974) Fracture resistance of paper. J Mater Sci 9(11):1745–1753

15. Shin C, Chung J, Kim W (2005) Dynamic characteristics of the out-of-plane vibration for an axially moving membrane. J Sound Vib 286(4–5):1019–1031
16. Simpson A (1973) Transverse modes and frequencies of beams translating between fixed end supports. J Mech Eng Sci 15(3):159–164
17. Swinehart D, Broek D (1995) Tenacity and fracture toughness of paper and board. J Pulp Pap Sci 21(11):J389–J397
18. Swope RD, Ames WF (1963) Vibrations of a moving threadline. J Franklin Inst 275(1):36–55
19. Ulsoy AG, Mote CD (1982) Vibration of wide band saw blades. J Eng Ind 104(1):71–78
20. Wang Y, Huang L, Liu X (2005) Eigenvalue and stability analysis for transverse vibrations of axially moving strings based on Hamiltonian dynamics. Acta Mech Sin 21(5):485–494
21. Westergaard HM (1939) Bearing pressures and cracks. J Appl Mech 6:A49–A53
22. Wickert JA, Mote CD (1990) Classical vibration analysis of axially moving continua. J Appl Mech 57(3):738–744

Chapter 22
Dynamic Behaviour of a Travelling Viscoelastic Band in Contact with Rollers

Tytti Saksa, Nikolay Banichuk, Juha Jeronen, Matti Kurki, and Tero Tuovinen

Abstract The dynamic behaviour of an axially moving viscoelastic band, in contact with supporting rollers, is studied. A model of a thin, viscoelastic beam (panel) subjected to bending and centrifugal forces is used. An initial-boundary value problem for a fifth-order partial differential equation describing the movement of the band is formulated in detail. In this paper, five boundary conditions in total are used for the first time within the present model. An external force describing the normal force of the roller supports is included. Combining this viscoelastic model with the roller contact simulation is a new approach among moving band behaviour studies. The initial-boundary value problem is solved numerically using the fourth-order Runge-Kutta method and the central finite differences, and the band behaviour is illustrated for different band velocities and degrees of viscosity. It is found that the damping effect of viscoelasticity increases when the band velocity increases, and that the roller contact has a greater effect on the elastic panel behaviour than on the viscoelastic panel behaviour.

T. Saksa (✉) · J. Jeronen · T. Tuovinen
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland
e-mail: tytti.saksa@jyu.fi

J. Jeronen
e-mail: juha.jeronen@jyu.fi

T. Tuovinen
e-mail: tero.tuovinen@jyu.fi

N. Banichuk
Ishlinsky Institute for Problems in Mechanics, Russian Academy of Sciences (RAS), Prospekt Vernadskogo 101, 119526 Moscow, Russia
e-mail: banichuk@ipmnet.ru

M. Kurki
School of Technology, JAMK University of Applied Sciences, P.O. Box 207, 40101 Jyväskylä, Finland
e-mail: matti.kurki@jamk.fi

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_22,

22.1 Introduction

The behaviour of systems, in which some material travels axially at a fast speed between two supports, has been studied widely. Interest in these studies arises from the extensive amount of applications in industry, e.g., in paper making processes.

In paper machines, the radius of supporting rollers is usually large compared to the length of an open draw, see Fig. 22.1. However, in the often studied models, the effect of the rollers on the behaviour of the moving web has been neglected.

Vibrations of travelling strings, beams, and bands were first studied by Sack [28], Archibald and Emslie [1], Miranker [22], Swope and Ames [30], Mote [24–26], Simpson [29], Ulsoy and Mote [31], Chonan [9], and Wickert and Mote [33]. These studies focused on one-dimensional free and forced vibrations including the nature of wave propagation in moving media and the effects of axial motion on the eigenfrequencies and eigenmodes. Stability of travelling two-dimensional rectangular membranes and plates was studied, e.g., by Ulsoy and Mote [32], Lin and Mote [19], Lin [18], and Banichuk et al. [2].

Archibald and Emslie [1] and Simpson [29] studied effects of the axial motion on the eigenfrequencies and eigenfunctions. It was shown that the natural frequency of each mode decreases when the transport speed increases, and that both the travelling string and beam experience divergence instability at a sufficiently high speed.

Wet paper and many other materials have viscoelastic properties. The first study on transverse vibrations of a travelling viscoelastic material was carried out by Fung et al. [11], who used a string model. They investigated numerically the effects of material parameters and transport velocity on the transient amplitudes. Extending their work, they studied the material damping effect in their later research [12]. Fung et al. used a standard linear solid model to describe the viscoelasticity of material.

String and beam models have been widely used models in the studies concerning travelling viscoelastic materials. Oh et al. [27] and Lee and Oh [17] studied critical speeds, eigenvalues, and natural modes of axially moving viscoelastic beams using a spectral element model.

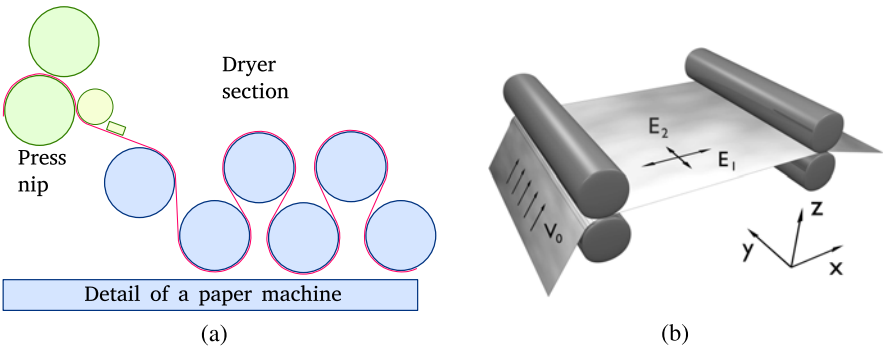


Fig. 22.1 An overview. (**a**) Paper machine cross-section. (**b**) A qualitative drawing of an open draw

Chen and Zhao [8] represented a modified finite difference method to simplify a non-linear model of an axially moving string. They studied numerically the free transverse vibrations of elastic and viscoelastic strings.

Yang and Chen [7, 34] studied vibrations and stability of axially moving viscoelastic beams with periodic parametric excitations. Yang and Chen [34] studied dynamic stability of axially moving viscoelastic beams with a time-pulsating speed. They found that the viscoelastic damping decreases the instability region of subharmonic resonance. Chen and Yang [7] studied free vibrations of a viscoelastic beam travelling between simple supports with torsion strings. They studied the viscoelastic effect by perturbing the similar elastic problem and using the method of multiple scales.

Marynowski and Kapitaniak [20] studied the difference between the Kelvin-Voigt model and the Bürgers model in internal damping and found out that both models give accurate results with a small damping coefficient, but with a large damping coefficient, the Bürgers model is more accurate. In 2007, they compared the models with the Zener model studying the dynamic behavior of an axially moving viscoelastic beam [21]. They found out that the Bürgers and Zener models gave similar results for the critical transport speed whereas the Kelvin-Voigt model gave significantly greater transport speed compared to the other two models.

In all discussed studies above, a partial time derivative has been used instead of a material derivative in the viscoelastic constitutive relations. Mockensturm and Guo [23] suggested that the material derivative should be used. They studied nonlinear vibrations and the dynamic response of axially moving viscoelastic strings, and found significant discrepancy in the frequencies at which non-trivial limit cycles exist comparing the models with the partial time derivative and the material time derivative. In Chen et al. [4], Ding and Chen [10], Chen and Wang [6], and Chen and Ding [5], the material derivative was also used in the viscoelastic constitutive relations. Ding and Chen [10] studied stability of axially accelerating viscoelastic beams using the method of multiple scales and parametric resonance. Chen and Wang [6] studied the stability of axially accelerating viscoelastic beams using the asymptotic perturbation analysis. In a recent research by Chen and Ding [5], the steady-state response of transverse vibrations for axially moving viscoelastic beams was studied. Kurki and Lehtinen [16] suggested, separately, that the material derivative in the constitutive relations should be used in their study concerning the in-plane displacement field of a travelling viscoelastic plate.

Using the material derivative in the viscoelastic constitutive relations for a beam model leads to a partial differential equation that is fifth-order with respect to the space coordinate. In Ding and Chen [10], Chen and Wang [6], and Chen and Ding [5], the fifth-order dynamic equation is attained but only four boundary conditions (in space) are used. However, the amount of boundary (initial) conditions should coincide with the order of the equation with respect to each variable.

We also mention studies by Guan (et al.) [13–15]. They used a different (from the references mentioned above) kind of approach in modelling of viscoelastic effects in moving web-handling systems applying the White–Metzner rheological equation. In those studies, permanent web deformations and web tension behaviour as a function of time were investigated.

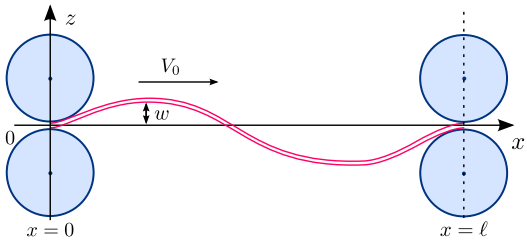


Fig. 22.2 An assumption of cylindrical deformation

In this study, we investigate the transverse displacement of a viscoelastic panel travelling between and in contact with two supports. Using a linear Kirchhoff plate model and a Kelvin-Voigt viscoelasticity model, a fifth-order partial differential equation for the transverse displacement of the panel is derived in detail. Simply supported boundary conditions are used at both edges and, at the in-flow edge, an additional boundary condition corresponding to the travelling angle is used. That is, five boundary conditions in total are used. The contact with the supporting rollers is modelled by a nonlinear spring force between the rollers and the panel. Numerical simulations of the behaviour of the panel are presented. A comparison of the behaviour between the model including the contact effect and the classic model with no contact is made.

22.2 Problem Setup

Consider a viscoelastic band travelling at a constant axial velocity V_0 (in the x direction) in a span. The domain of this study is the span between two rollers located at $x = 0$ and $x = \ell$. We investigate the transverse displacement w of the band as a dynamic problem taking into account the contact with the rollers. We assume that the transverse displacements are small to make the linear theory justifiable. We also assume that the displacement w is cylindrical, that is, the displacement does not vary in the cross direction to the movement, see Fig. 22.2. The thickness of the band is assumed to be constant, h. The tension at the edges is supposed to be constant, T_0. The plate is assumed to have a constant bending rigidity, D, and a constant viscous bending rigidity D^{v}. The mass per area of the band is m.

The equation describing the transverse displacement $w = w(x, t)$ of the panel (a plate with cylindrical deformation) is derived using the Kirchhoff plate model and the Kelvin-Voigt model for the viscoelasticity.

We first write the equilibrium equation for the bending forces affecting the panel, which is

$$\frac{\partial^2 M}{\partial x^2} + T_0 \frac{\partial^2 w}{\partial x^2} + q = 0, \quad x \in (0, \ell), \tag{22.1}$$

where T_0 is the tension force in the x direction, M the bending moment, and q the intensity of external load distributed over the upper surface of the panel.

Let σ denote the flexural stress. This stress depends on a strain that is defined by

$$\varepsilon = -z\,\frac{\partial^2 w}{\partial x^2}. \tag{22.2}$$

The bending moment is related to the flexural stress by

$$M = \int_{-h/2}^{h/2} z\sigma \,\mathrm{d}z. \tag{22.3}$$

The stress depends on the strain by the relation

$$\sigma = C\varepsilon + \Gamma\frac{\mathrm{d}\varepsilon}{\mathrm{d}t}, \tag{22.4}$$

where

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\partial}{\partial t} + V_0\frac{\partial}{\partial x},$$

and

$$C = \frac{E}{1-\nu^2}, \qquad \Gamma = \frac{\eta}{1-\varphi^2}.$$

Here, E is the Young modulus, ν the Poisson ratio, and η and φ are the corresponding viscous material constants.

For the balance equation (22.1), we calculate, first, the bending moment. By inserting (22.4) into (22.3) and (22.2) into (22.4), we obtain

$$M = \int_{-h/2}^{h/2} z\sigma \,\mathrm{d}z = \int_{-h/2}^{h/2} z\left(C\varepsilon + \Gamma\frac{\mathrm{d}\varepsilon}{\mathrm{d}t}\right)\mathrm{d}z = -\frac{h^3}{12}\left(C\,\frac{\partial^2 w}{\partial x^2} + \Gamma\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial^2 w}{\partial x^2}\right). \tag{22.5}$$

We calculate the second space derivative of the bending moment (22.5). We obtain

$$\frac{\partial^2 M}{\partial x^2} = -\frac{h^3}{12}\left(C\,\frac{\partial^4 w}{\partial x^4} + \Gamma\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial^4 w}{\partial x^4}\right). \tag{22.6}$$

Substituting (22.6) into (22.1), we obtain

$$-\frac{h^3}{12}\left(C\,\frac{\partial^4 w}{\partial x^4} + \Gamma\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial^4 w}{\partial x^4}\right) + T_0\,\frac{\partial^2 w}{\partial x^2} + q = 0. \tag{22.7}$$

Introducing the parameters

$$D = \frac{h^3}{12}\,C, \qquad D^{\mathrm{v}} = \frac{h^3}{12}\,\Gamma,$$

and adding dynamical components into (22.7), we obtain

$$-D\,\frac{\partial^4 w}{\partial x^4} - D^{\mathrm{v}}\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial^4 w}{\partial x^4} + T_0\,\frac{\partial^2 w}{\partial x^2} + q = m\,\frac{\mathrm{d}^2 w}{\mathrm{d}t^2}. \tag{22.8}$$

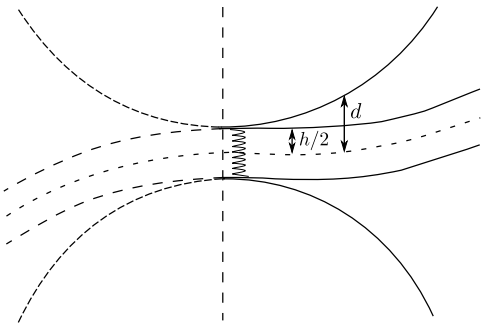


Fig. 22.3 A spring model in the cross direction of the plate. A detail near one end of the span

Expanding the expressions in (22.8) and re-organizing the terms, the dynamic equation for $w = w(x,t)$ reads

$$\frac{\partial^2 w}{\partial t^2} + \left(2V_0\frac{\partial}{\partial x} + \frac{D^{\mathrm{v}}}{m}\frac{\partial^4}{\partial x^4}\right)\frac{\partial w}{\partial t} + \left[\left(V_0^2 - \frac{T_0}{m}\right)\frac{\partial^2}{\partial x^2} + \frac{D}{m}\frac{\partial^4}{\partial x^4} + V_0\frac{D^{\mathrm{v}}}{m}\frac{\partial^5}{\partial x^5}\right]w = \frac{q}{m}, \tag{22.9}$$

where $x \in (0,\ell)$, $t \in (0,t_{\mathrm{f}})$, and t_{f} is the end point of the time domain. We use classical simply supported boundary conditions at both edges, and an additional condition at the in-flow edge. The boundary conditions read

$$w(0,t) = w(\ell,t) = 0, \qquad \frac{\partial^2 w}{\partial x^2}(0,t) = \frac{\partial^2 w}{\partial x^2}(\ell,t) = 0, \tag{22.10}$$

and

$$\frac{\partial w}{\partial x}(0,t) = \theta, \tag{22.11}$$

where θ is a given constant describing the angle between the panel and the x axis at the in-flow edge. The angle θ represents the feeding angle of the web, and in a multi-span system it could be predicted (calculated) for one span from the behaviour of the panel on the preceding span. However, in this study we concentrate only on one isolated span and assume that the feeding angle is known. The initial conditions for the dynamic problem are

$$w(x,0) = g_1(x), \qquad \frac{\partial w}{\partial t}(x,0) = g_2(x), \tag{22.12}$$

where g_1 and g_2 are some given functions.

The contact force between the moving panel and the supporting rollers is now to be included in the panel model. The transverse direction of the panel is modelled as a non-linear spring such that the maximum compression of the panel is one half of its thickness, see Figs. 22.3 and 22.4. The force function depending on the distance

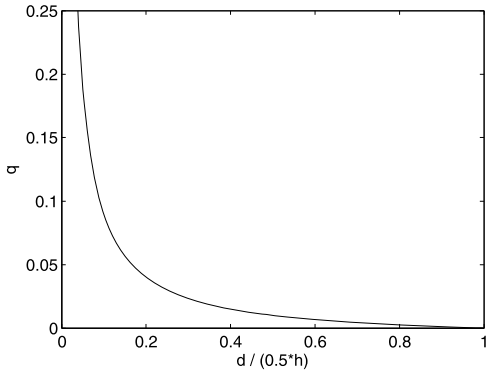


Fig. 22.4 Contact force as a function of the distance between the plate center and the roller

d between the panel center and the roller surface is given by

$$q(d) = a\left(\frac{h}{2d} - 1\right), \quad 0 < d \leq h/2. \tag{22.13}$$

The parameter a is a constant describing the strength of the force. Inside the rollers ($d \leq 0$) the force is not defined, and if there is no contact ($d > h/2$), then the force q is zero.

22.3 Numerical Investigation

We use central difference formulae and the fourth-order Runge-Kutta for the space and time discretisations, respectively. In the central differences, the higher order derivatives need node values from a distance of three nodes of the node being computed. We neglect the fifth-order derivatives at the boundary. The interval $[0, \ell]$ is divided to $n+1$ subintervals equal in length. The end points of the subintervals are labeled as $0 = x_0, x_1, \ldots, x_n, x_{n+1} = \ell$. We need one virtual point from the boundary conditions for both edges. From the boundary conditions, we get $w(x_0) = 0$, $w(x_{n+1}) = 0$, $w(x_{-1}) = -w(x_1)$ and $w(x_{n+2}) = -w(x_n)$. In boundary condition (22.11), we choose $\theta = 0$, which leads to $w(x_{-1}) = w(x_1)$ and finally $w(x_1) = 0$.

In Fig. 22.5, it is illustrated how the rollers and computation nodes are connected by simple geometry. It must be noticed that we are considering the transverse displacements merely, and therefore, the contact force effects are considered in the z direction only.

The parameters used are as follows:

$$\begin{aligned} &\ell = 0.25 \text{ m}, \quad && T_0 = 500 \text{ N/m}, \quad && h = 10^{-4} \text{ m}, \quad && m = 0.08 \text{ kg/m}^2, \\ &E = 10^9 \text{ Pa}, && \nu = 0.3, && h_s = 0.5 \cdot h, && r_s = 0.12 \text{ m}. \end{aligned} \tag{22.14}$$

Here, ℓ is the length of the open draw, T_0 is constant tension applied at the ends of the panel, h is the thickness of the panel, m is mass per unit area, E is the Young

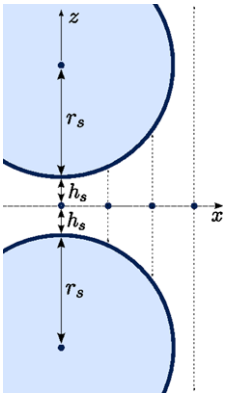


Fig. 22.5 Nodes between the rollers. A detail near one end of the span

modulus, ν is the Poisson ratio, r_s is the radius of the rollers, and h_s is one half of the distance between the pressing rollers, see Fig. 22.5. We define

$$D^{\mathrm{v}} = \alpha_{\mathrm{v}}\, D.$$

The multiplier α_{v} is here called the relative viscosity, for which the values $\alpha_{\mathrm{v}} = 0.0008, 0.08$ were used. We studied the dynamic behaviour of the panel for the first 0.05 seconds, for three different velocities $V_0 = 0,\ 30,\ 60$ m/s. The strength of the force (22.13) was $a = 0.01$. The used number of the computation nodes was $n = 150$.

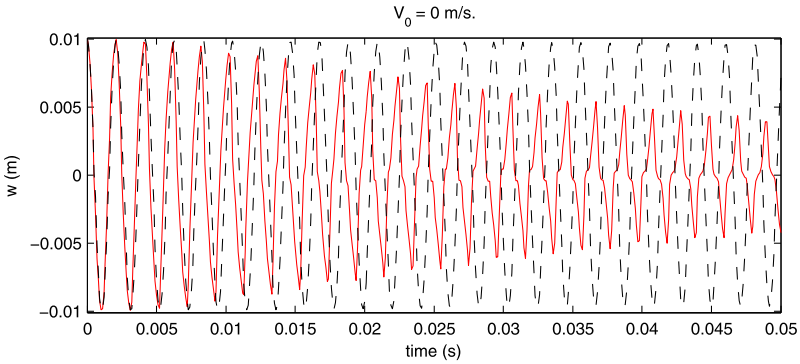
The used initial conditions were

$$w(x,0) = 0.01 \sin\left(\frac{\pi x}{\ell}\right),$$

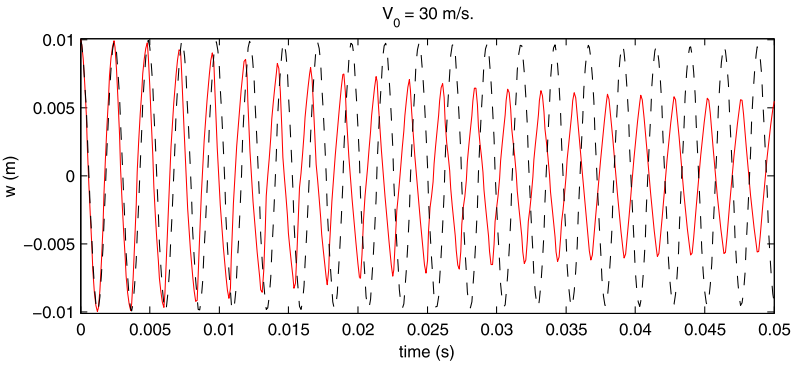
$$\frac{\partial w}{\partial t}(x,0) = 0.$$

The investigated cases include the behaviour of the midpoint of the panel (Figs. 22.6 and 22.7), from which the frequency and amplitude of the vibrations can be analysed, and the space-time behaviour of the panel (Figs. 22.8 and 22.10). The results for the stationary panel are shown in Figs. 22.6a, 22.8 (almost elastic material), and Figs. 22.7(a), 22.9 (viscoelastic material). The results for the moving panel are shown in Figs. 22.6(b), 22.6(c), 22.10 (almost elastic material), and Figs. 22.7(b), 22.7(c), 22.11 (viscoelastic material).

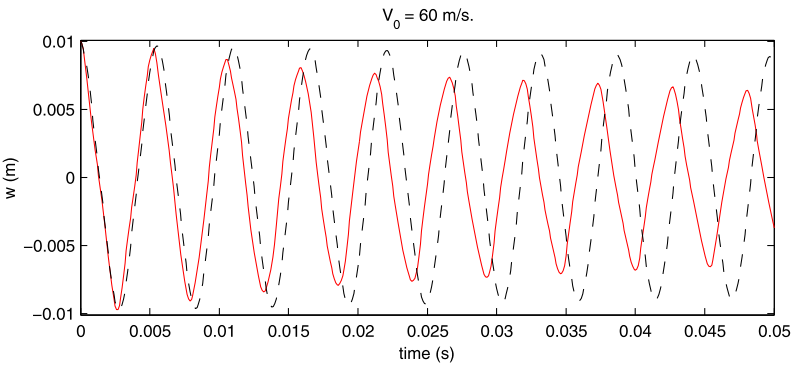
In Figs. 22.6 and 22.7, the behaviour of the panel center is shown for a panel travelling at different velocities for both viscoelastic and almost elastic materials. From Figs. 22.6(a) and 22.7(a), it can be seen that the contact with the rollers is decreasing the amplitude of the vibrations in the case of an almost elastic panel and increasing the amplitude in the case of a viscoelastic panel compared to the case with no roller contact. In both cases, the frequency of the vibrations is increased. Also, when the panel is moving at a constant velocity (Figs. 22.6(b), 22.6(c), 22.7(b), and 22.7(c)), the frequency of vibrations in the case with roller contact is greater



(a) $V_0 = 0$, $\alpha_v = 0.0008$.

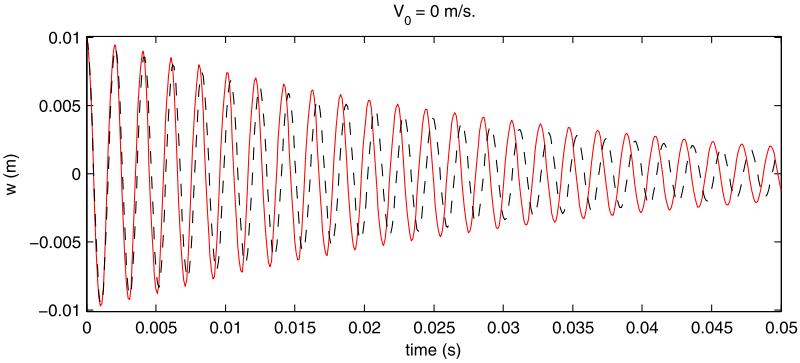


(b) $V_0 = 30$ m/s, $\alpha_v = 0.0008$.

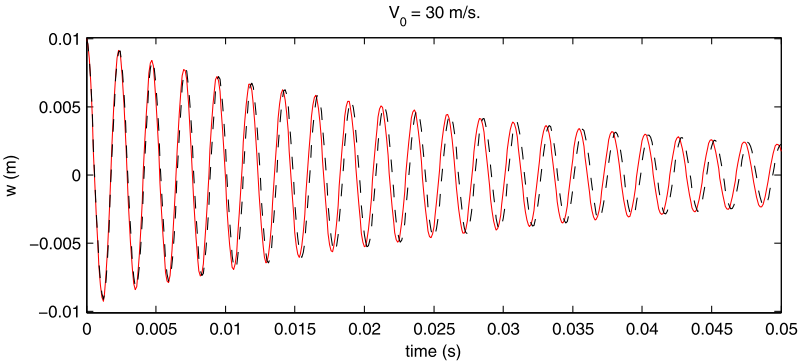


(c) $V_0 = 60$ m/s, $\alpha_v = 0.0008$.

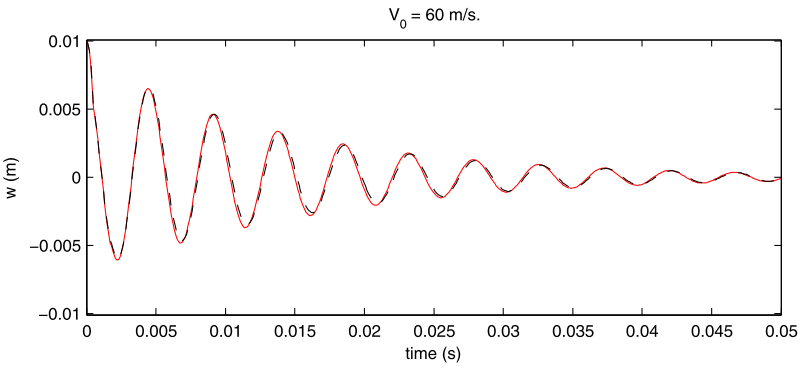
Fig. 22.6 Behaviour of the midpoint of the panel during the first 0.05 seconds for an almost elastic material. The *solid line* shows the case with roller contact, and the *dashed line* shows the case without contact. V_0 is the panel velocity and α_v is the relative viscosity



(a) $V_0 = 0$, $\alpha_v = 0.08$.

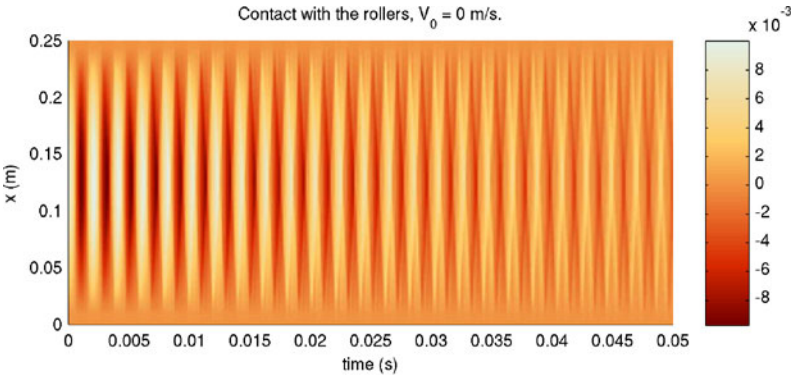


(b) $V_0 = 30$ m/s, $\alpha_v = 0.08$.

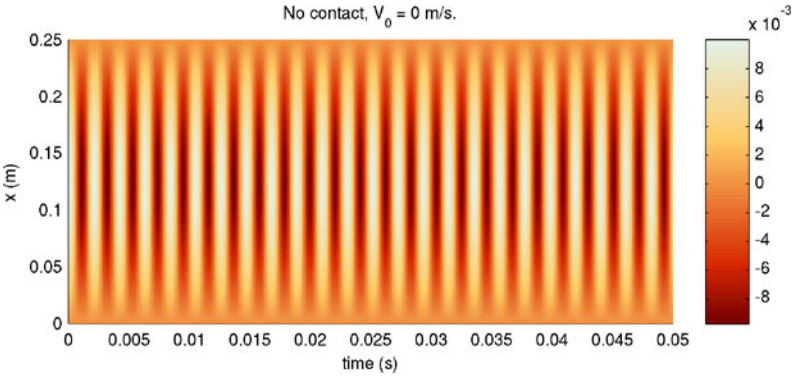


(c) $V_0 = 60$ m/s, $\alpha_v = 0.08$.

Fig. 22.7 Behaviour of the midpoint of the panel during the first 0.05 seconds for a viscoelastic material. *Solid line* shows the case with roller contact, and the *dashed line* shows the case without contact. V_0 is the panel velocity and α_v is the relative viscosity



(a) Contact, $\alpha_v = 0.0008$.



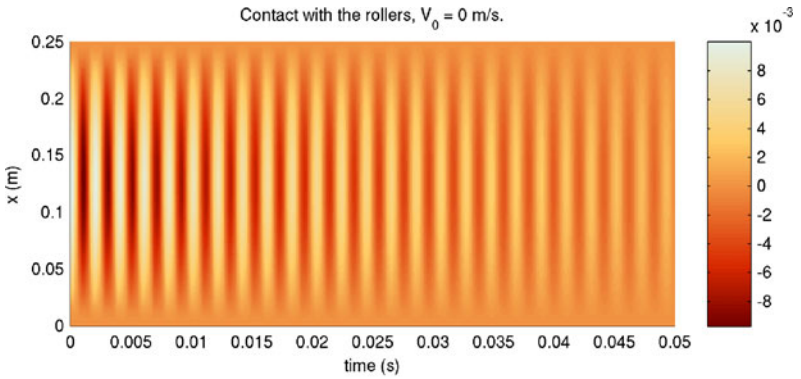
(b) No contact, $\alpha_v = 0.0008$.

Fig. 22.8 Behaviour of the panel during the first 0.05 seconds, when the panel is not moving ($V_0 = 0$). Almost elastic material, $\alpha_v = 0.0008$

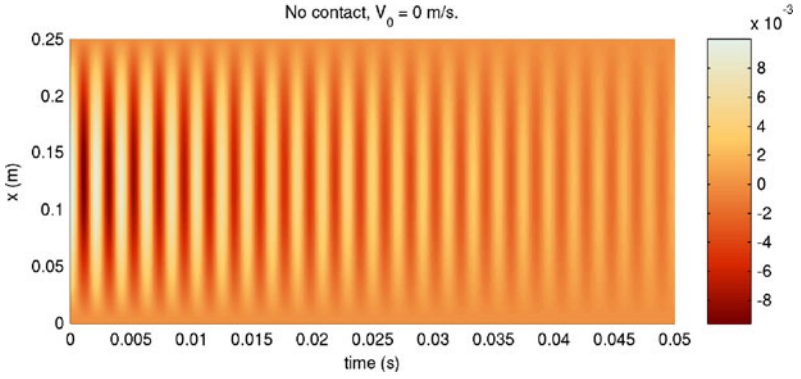
to the case with no roller contact. When the viscoelastic panel is moving fast (see Fig. 22.7(c)), the viscous damping is so fast that the effect of contact cannot be noticed.

In Figs. 22.8, 22.9, 22.10 and 22.11, colourshеets of the panel behaviour are provided for different panel velocities ($V_0 = 0, 30, 60$ m/s) for both almost elastic and viscoelastic materials. For a stationary panel, also the cases with no contact are drawn as reference cases (Figs. 22.8(b) and 22.9(b)). It can be seen that the viscous damping depends on the panel velocity and the relative viscosity. For a stationary panel (Figs. 22.8 and 22.9), the effect of the contact can be seen clearly for the almost elastic panel but the effect is very slight for the viscoelastic panel.

For a moving panel (Figs. 22.10 and 22.11), it can be noted that the upper-x half of the panel experiences its maximum or minimum amplitude before the lower-x half does. Similar behaviour was reported in [3]. The effect of viscous damping increases if the panel velocity is increased. Note that the viscoelastic panel (beam)



(a) Contact, $\alpha_v = 0.08$.



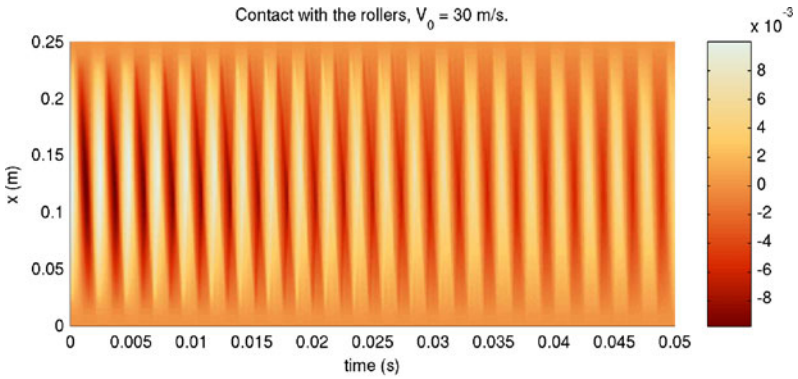
(b) No contact, $\alpha_v = 0.08$.

Fig. 22.9 Behaviour of the panel during the first 0.05 seconds, when the panel is not moving ($V_0 = 0$). Viscoelastic material, $\alpha_v = 0.08$

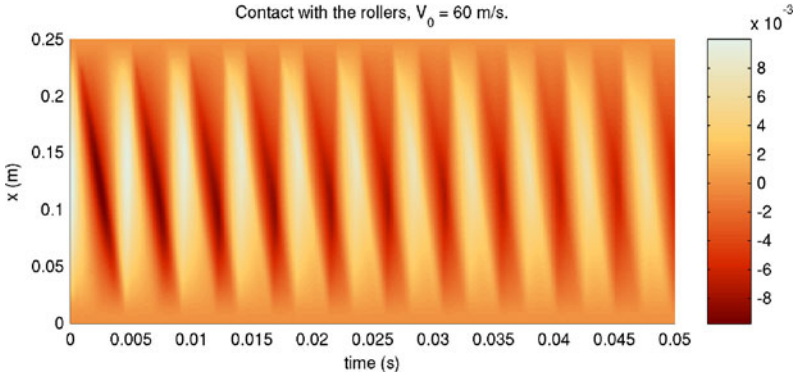
is expected to experience divergence instability at a sufficiently high speed, and the divergence velocity is expected to be close to the one of an elastic panel (beam) [17]. The critical velocity of a travelling elastic panel can be determined analytically by $(V_0)_{cr} = \sqrt{T_0/m + (\pi/\ell)^2 D/m} \approx 79.0581$ m/s [33].

22.4 Conclusions

In this study, the dynamical behaviour of an axially moving viscoelastic panel in contact with supporting rollers was investigated. The combination of the contact model with this kind of viscoelastic model was done for the first time. The dynamical equation describing the panel vibrations was derived and an initial-boundary value problem was formulated. The continuum equation was discretised via central finite differences in space and by the fourth-order Runge-Kutta method in time,



(a) Velocity $V_0 = 30$ m/s, $\alpha_v = 0.0008$.



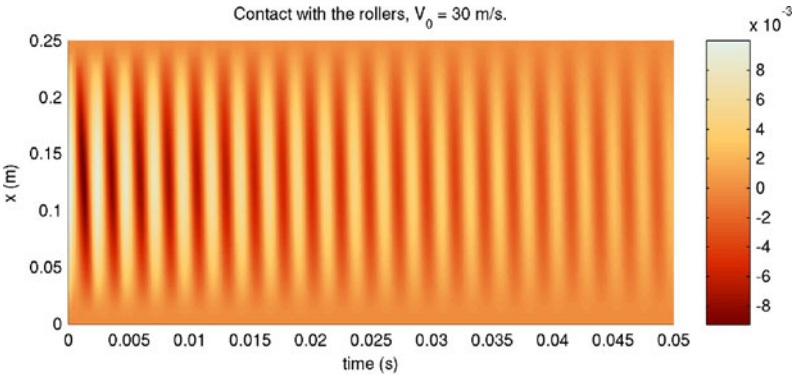
(b) Velocity $V_0 = 60$ m/s, $\alpha_v = 0.0008$.

Fig. 22.10 Behaviour of the panel during the first 0.05 seconds, when the panel is moving at a constant velocity. Almost elastic material, $\alpha_v = 0.0008$

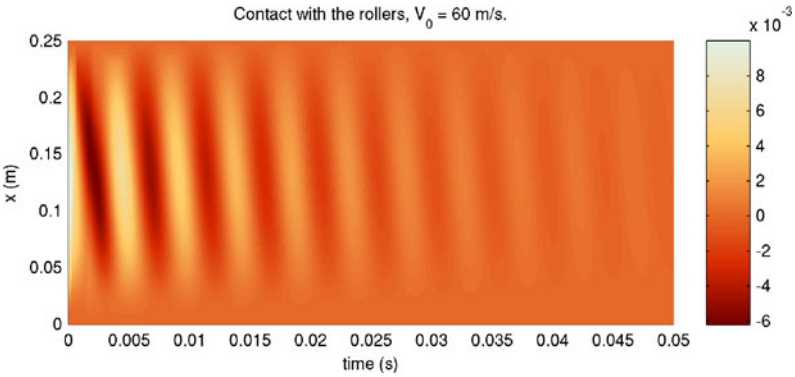
and solved numerically. Dynamics of the panel was studied for different relative viscosities and for different panel velocities, and the effect of roller contact was investigated by comparing the behaviour including contact with the behaviour with no contact.

In this study, it was noted that in the partial differential equation, describing the dynamics of an viscoelastic panel or beam and which is fifth-order in space, the amount of boundary conditions must be five in total.

From numerical investigations, it was seen that the contact force may decrease the amplitude of vibrations in the case of an almost elastic panel and increase the amplitude in the case of a viscoelastic panel compared to the case with no roller contact. The decrease in viscous damping introduced by the roller contact was surprising. It was also noted that the viscous damping increases a lot when the panel velocity is increased.



(a) Velocity $V_0 = 30$ m/s, $\alpha_v = 0.08$.



(b) Velocity $V_0 = 60$ m/s, $\alpha_v = 0.08$.

Fig. 22.11 Behaviour of the panel during the first 0.05 seconds, when the panel is moving at a constant velocity. Viscoelastic material, $\alpha_v = 0.08$

Note that, in this study concerning moving viscoelastic panels, the effects of surrounding fluid were excluded to investigate solely the role of material viscoelasticity in the panel dynamics. The presence of fluid is known to considerably affect the panel behaviour [3], and thus the present study should primarily be seen as academic basic research. The behaviour of a moving viscoelastic panel submerged in fluid remains a topic of future research.

Acknowledgements This research has been supported by the Jenny and Antti Wihuri foundation, and the Academy of Finland (grant no. 140221).

References

1. Archibald FR, Emslie AG (1958) The vibration of a string having a uniform motion along its length. J Appl Mech 25:347–348

2. Banichuk N, Jeronen J, Kurki M, Neittaanmäki P, Saksa T, Tuovinen T (2011) On the limit velocity and buckling phenomena of axially moving orthotropic membranes and plates. Int J Solids Struct 48(13):2015–2025
3. Banichuk N, Jeronen J, Neittaanmäki P, Tuovinen T (2011) Dynamic behaviour of an axially moving plate undergoing small cylindrical deformation submerged in axially flowing ideal fluid. J Fluids Struct 27(7):986–1005
4. Chen L-Q, Chen H, Lim CW (2008) Asymptotic analysis of axially accelerating viscoelastic strings. Int J Eng Sci 46(10):976–985. doi:10.1016/j.ijengsci.2008.03.009
5. Chen L-Q, Ding H (2010) Steady-state transverse response in coupled planar vibration of axially moving viscoelastic beams. J Vib Acoust 132:011009. doi:10.1115/1.4000468
6. Chen L-Q, Wang B (2009) Stability of axially accelerating viscoelastic beams: asymptotic perturbation analysis and differential quadrature validation. Eur J Mech A, Solids 28(4):786–791. doi:10.1016/j.euromechsol.2008.12.002
7. Chen L-Q, Yang X-D (2006) Vibration and stability of an axially moving viscoelastic beam with hybrid supports. Eur J Mech A, Solids 25(6):996–1008. doi:10.1016/j.euromechsol.2005.11.010
8. Chen L-Q, Zhao W-J (2005) A numerical method for simulating transverse vibrations of an axially moving string. Appl Math Comput 160(2):411–422. doi:10.1016/j.amc.2003.11.012
9. Chonan S (1986) Steady state response of an axially moving strip subjected to a stationary lateral load. J Sound Vib 107:155–165
10. Ding H, Chen L-Q (2008) Stability of axially accelerating viscoelastic beams: multi-scale analysis with numerical confirmations. Eur J Mech A, Solids 27(6):1108–1120. doi:10.1016/j.euromechsol.2007.11.014
11. Fung R-F, Huang J-S, Chen Y-C (1997) The transient amplitude of the viscoelastic travelling string: an integral constitutive law. J Sound Vib 201(2):153–167. doi:10.1006/jsvi.1996.0776
12. Fung R-F, Huang J-S, Chen Y-C, Yao C-M (1998) Nonlinear dynamic analysis of the viscoelastic string with a harmonically varying transport speed. Comput Struct 66(6):777–784. doi:10.1016/S0045-7949(98)00001-7
13. Guan X (1995) Modeling of viscoelastic effects on web handling system behavior. PhD thesis, Oklahoma State University
14. Guan X, High MS, Tree DA (1995) Viscoelastic effects in modeling web handling systems: steady-state analysis. J Appl Mech 62(4):908–914. doi:10.1115/1.2789031
15. Guan X, High MS, Tree DA (1998) Viscoelastic effects in modeling web handling systems: unsteady-state analysis. J Appl Mech 65(1):234–241. doi:10.1115/1.2789031
16. Kurki M, Lehtinen A (2009) In-plane strain field theory for 2-d moving viscoelastic webs. In: Papermaking research symposium 2009, Kuopio, Finland. PRS
17. Lee U, Oh H (2005) Dynamics of an axially moving viscoelastic beam subject to axial tension. Int J Solids Struct 42(8):2381–2398
18. Lin CC (1997) Stability and vibration characteristics of axially moving plates. Int J Solids Struct 34(24):3179–3190
19. Lin CC, Mote CD (1995) Equilibrium displacement and stress distribution in a two-dimensional, axially moving web under transverse loading. J Appl Mech 62:772–779
20. Marynowski K, Kapitaniak T (2002) Kelvin-Voigt versus Bürgers internal damping in modeling of axially moving viscoelastic web. Int J Non-Linear Mech 37(7):1147–1161. doi:10.1016/S0020-7462(01)00142-1
21. Marynowski K, Kapitaniak T (2007) Zener internal damping in modelling of axially moving viscoelastic beam with time-dependent tension. Int J Non-Linear Mech 42(1):118–131. doi:10.1016/j.ijnonlinmec.2006.09.006
22. Miranker WL (1960) The wave equation in a medium in motion. IBM J Res Dev 4:36–42
23. Mockensturm EM, Guo J (2005) Nonlinear vibration of parametrically excited, viscoelastic, axially moving strings. J Appl Mech 72(3):374–380. doi:10.1115/1.1827248
24. Mote CD (1968) Divergence buckling of an edge-loaded axially moving band. Int J Mech Sci 10:281–295
25. Mote CD (1972) Dynamic stability of axially moving materials. Shock Vib Dig 4(4):2–11

26. Mote CD (1975) Stability of systems transporting accelerating axially moving materials. J Dyn Syst Meas Control 97:96–98
27. Oh H, Cho J, Lee U (2004) Spectral element analysis for an axially moving viscoelastic beam. J Mech Sci Technol 18(7):1159–1168. doi:10.1007/BF02983290
28. Sack RA (1954) Transverse oscillations in traveling strings. Br J Appl Phys 5:224–226
29. Simpson A (1973) Transverse modes and frequencies of beams translating between fixed end supports. J Mech Eng Sci 15:159–164
30. Swope RD, Ames WF (1963) Vibrations of a moving threadline. J Franklin Inst 275:36–55
31. Ulsoy AG, Mote CD (1980) Analysis of bandsaw vibration. Wood Sci 13:1–10
32. Ulsoy AG, Mote CD (1982) Vibration of wide band saw blades. J Eng Ind 104:71–78
33. Wickert JA, Mote CD (1990) Classical vibration analysis of axially moving continua. J Appl Mech 57:738–744
34. Yang X-D, Chen L-Q (2005) Dynamic stability of axially moving viscoelastic beams with pulsating speed. Appl Math Mech 26(8):905–910

Chapter 23
Visual Contrast Preserving Representation of High Dynamic Range Mathematical Functions

Juha Jeronen

Abstract When Gaussian distributed inputs, representing model parameters with some measurement error, are mapped through certain mechanical vibration models, the corresponding output probability distribution exhibits an approximately logarithmic data value distribution (in the histogram sense) with a high dynamic range (HDR). We look at applying tone mapping techniques from HDR photography to produce a low dynamic range, visual contrast preserving representation of such high dynamic range mathematical functions—thus enabling HDR plotting. This makes it possible to visualize HDR functions, displaying their structure in a clear manner on standard low dynamic range media such as computer screens and print. The advantages over simple logarithmic scaling are the visual contrast preservation and data adaptivity. Comparing to histogram equalization, the present approach has the advantage of not exaggerating small contrasts. Three methods are suggested and demonstrated on two mechanical vibration problems: transverse waves in a classical vibrating string, and the dynamic out-of-plane behaviour of an axially travelling panel submerged in axial potential flow.

23.1 Introduction

What are high dynamic range mathematical functions, and where would one want to visualize them? The motivation for this study comes from physics and engineering problems where model input is never exact. To obtain reliable analysis results, it is desirable to find out how stable the predictions of a given model are with respect to small perturbations in model input, and how large the expected range of output is. When input uncertainties are present, instead of a single solution, one obtains a solution set corresponding to the admissible inputs.

Adopting a direct statistical approach to uncertainty analysis, it is possible, in the case of computationally lightweight models, to approximate the solution set

J. Jeronen (✉)
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland
e-mail: juha.jeronen@jyu.fi

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_23,

directly. This is done via sampling the multidimensional input probability distribution (corresponding to the set of admissible parameter combinations when the uncertainty is accounted for; Latin hypercube techniques can be used, see, e.g., [5, 23, 24, 33, 34, 46]), and mapping the samples through the model being analyzed. Different statistical quantities can then be computed based on the discrete output sample. See, e.g., [25] for a general review of statistical uncertainty and sensitivity analysis techniques.

In order to facilitate an intuitive understanding of the results, it is also possible to visualize the output probability density directly.[1] The density of the resulting output probability distribution is estimated from the discrete output sample (using, for example, kernel density estimation; see, e.g., [7, 8, 16, 43]), and the resulting probability density field is then plotted. See Figs. 23.7–23.9 below for examples.

Applying this methodology to mechanical vibration problems, it was found that the resulting probability field does not lend itself to traditional linear scaling for visualization. The reason is that the histogram of the data is distributed, approximately, in a logarithmic manner, and the dynamic range spanned by the data far exceeds the representable range of a computer display: it is a high dynamic range mathematical function.

Traditionally, logarithmically distributed data with a high dynamic range occurs in contexts such as audio and high dynamic range (HDR) photography. Thus, considering the task of visualizing a HDR function, it is natural to seek methods for representation of HDR data that have been developed in these fields.

The need for methods to represent HDR data in both of these classical fields is clear. The human eye has a range of five orders of magnitude in light intensity (e.g., [12]), while computer displays (in terms of photometric light intensity) are limited to two. Similarly, standard digital audio is sampled at 16 bits per sample, which gives a range of about four orders of magnitude (logarithmically, 90 dB).

In audio processing, the standard solution is to work on the decibel scale, which is logarithmic, allowing an at-a-glance representation of the HDR data. However, in HDR photography literature, it is well known that simple logarithmic scaling has a tendency to eliminate contrast (e.g., [28]).

For the purposes of HDR photography, special *tone mapping* algorithms have been developed for the purpose of representing, in a visually accurate manner, high dynamic range scenes on low dynamic range media such as regular computer displays and print. The basic idea behind tone mapping is that the human visual system is sensitive to differences in light intensity, but not to absolute intensities [28]. In terms of signal processing, tone mapping can be seen as data-adaptive dynamic range compression. As tone mapping algorithms are important for the focus of our study, we will review the related literature in the next section.

The application of tone mapping techniques for plotting HDR mathematical functions can be seen as a natural extension of the ideas of Park and Montag [35],

[1] SAVU, **S**ample-based **A**nalysis and **V**isualization of **U**ncertainty: https://yousource.it.jyu.fi/savu/codes/ Link cited 13 Jan 2012.

who investigated the use of tone mapping for representation of data from astronomical and medical imaging, captured at wavelengths other than those of visible light.

There is a large body of research in HDR signal processing, which is not limited to tone mapping only. Some examples follow. Display adaptation in different ambient lighting conditions is discussed in [30], and feature classification in areas obscured by shadows in [9]. Adapting HDR images to target devices with drastically different dynamic ranges is considered in [50].

The study [11] concentrates on the problem of obtaining HDR radiance maps by stitching together multiple LDR (i.e. the usual kind of) photographs taken with different exposure parameters; this is similar to stitching together a panorama, but along the light intensity axis. The same problem has been discussed in [29] earlier, and [1] is a recent technical report on the subject. Integration of computer-generated (3D rendered) objects into HDR photographs has been investigated in [10].

23.2 Tone Reproduction Operators and HDR Plotting

Considering visual contrast preserving representation of high dynamic range mathematical functions, it seems that the task has received very little attention. Some partly relevant studies exist; for example, [51] discusses volumetric visualization of HDR data. The study [36] talks about the potentially deceptive appearance of logarithmic scaling for data obtained from cytometry, and suggests a new scaling method for that particular application.

Most interestingly, in [35] a psychophysical study was carried out on applying different tone reproduction operators (TROs, see below) on HDR data from, e.g., medical, astronomical, and radar sources. The authors conclude that aside from some general trends, the appropriate tone reproduction operator ultimately depends on the kind of data, and on expert opinion in the specific field of application. The study concentrates mainly on qualitative aspects of the user experience, and does not consider extension into visualization of mathematical functions.

Tone reproduction operators (TROs) are used in the conversion of HDR images for display on standard dynamic range (low dynamic range; LDR) devices and media, which include regular computer displays and print. The conversion is known as *tone mapping*. TROs scale the data in an adaptive manner to maintain visibility of detail, and sometimes also simulate aspects of the human visual system.

The methods can be broadly divided into two categories. There are global methods (such as [14, 15, 28]; see also [12, 39, 42] for more references), which apply the same mapping function to each pixel in the image, and local methods (e.g., [2, 17, 18, 22]), which may vary the mapping across the image. Global methods operate on the histogram of the image, often on a logarithmic intensity scale, while local methods operate directly on the image data.

The problems with classical scalings, motivating the creation of TROs, are as follows. If a simplistic linear scaling and quantization procedure is used, often a large portion of the light intensity data falls into the first few bins (e.g., [35, 39] note this,

providing examples; see also Fig. 23.2 below). Scaling the logarithm of the intensity linearly and quantizing the result (i.e. plotting on a logarithmic scale), on the other hand, eliminates contrast [28]. Finally, the technique of histogram equalization makes the density histogram constant, which not only compresses large contrasts, but also exaggerates contrast in sparsely populated parts of the histogram ([28]; see also our example in Fig. 23.2 below). Histogram equalization may serve better in preprocessing input for pattern recognition in machine vision, as suggested in [19].

In the paper [39], a large number of different local and global TROs are reviewed and tested, and a new one is proposed. The authors suggest that most TROs, regardless of whether they are global or local, can be approximated to a satisfying degree by simple, fast image processing operations; this observation is also made in [50]. In the paper [31], the observation is tested quantitatively by approximating a number of different operators via a simplified model and parameter fitting. The authors suggest their model as an approach for validation and comparison of TROs.

Regardless of whether a global or a local tone mapping method is used, several authors caution against gradient reversal, i.e. flips of the local gradient direction in the dynamics compressed LDR image when compared to the original HDR data. Care must be taken because gradient reversal may create dark halos around bright objects. It can be avoided by careful design of the TRO. See, e.g., [18, 28, 47].

The study [41] discusses both global and local methods for tone mapping, and notes the globally order-preserving property of some global TROs; mathematically, it is obvious that the required property is the monotonicity of the mapping function. Specifically for photographs, [41] notes that, due to the well-known perceptual illusion that may make the same intensity look different when surrounded by brighter or darker shades, local tone-mapping techniques are preferable.

For the purposes of HDR plotting, it is clear that a global method is more appropriate. This is because then (right until quantization into pixel values) there is a one-to-one mapping between the original data value and the dynamics compressed function value.[2] Hence, a global colour bar (or, e.g., a global vertical scale, if the y axis is compressed instead of pixel intensity) can be made; this would be impossible for a local method.

In the study [47], it is cautioned that even though by using global techniques one can easily avoid gradient reversals, they may cause reversals in gradient magnitudes. An originally small difference between two pixels falling into a densely populated part of the histogram may look larger in the resulting LDR image than an originally large difference between two pixels that fall into a sparsely populated part. This is because the TRO may attenuate more aggressively in the sparsely populated parts of the histogram, in order to make more room on the output intensity scale for the densely populated parts. This may happen even if the method prevents contrast expansion, since what matters here is the ratio of contrast attenuation factors

[2] Strictly speaking, in the case of data-adaptive histogram remappers, if some of the histogram bins are empty, there may be a flat region in the mapping function. In this case the mapping is not globally one-to-one. However, since such regions contain no samples in the data, for the existing data it is one-to-one.

in different parts of the histogram. Local methods can work around the problem by locally adapting the mapping; however, as was noted above, this is of little use for HDR plotting. In the present study, we have chosen to ignore this problem.

It should be noted that in HDR-to-LDR conversion of photographic images in particular, various kinds of perception models of the human visual system are commonly used. Perceptual models account for factors such as gradual loss of colour sensitivity at low intensities [27, 28], loss of visual acuity at low intensities [27, 28], veiling glare [27, 28, 32, 45], global adaptation of vision to the intensity at the foveal point (e.g., [47]), and time-dependent adaptation [14, 21, 37], [48, Refs. 15–21]. See also, e.g., [13, 20, 44]. In the present study, we will not consider perceptual modelling.

The review [12] presents a comprehensive and quickly readable overview and classification of different kinds of TROs up to the year 2002. In 2010, global TROs specifically were reviewed in [42]. A psychophysical study comparing user preferences for different TROs in the photographic context was carried out in [49]. The 2007 paper [40] reviews TROs, and raises an important point about the input and output domains of the operators. The authors argue that operators based on perceptual models (unlike ones based on engineering principles such as histogram remappers) require both forward (light intensity to luminance) and backward (luminance to light intensity) passes to produce valid results. In the present study, we will concentrate on histogram remappers only; hence a single pass is sufficient.

23.3 The Dynamic Range

For grayscale (scalar) data, the representable dynamic range of a given medium is defined as (e.g., [41])

$$B' = \frac{|d_{\max}|}{|d_{\min}|}, \tag{23.1}$$

where B' denotes the dynamic range, and $|d_{\max}|$ and $|d_{\min}|$ refer to the largest and smallest nonzero representable data values (in terms of absolute value), respectively.

It is convenient to use a logarithmic representation. On the decibel scale, we have the equivalent definition

$$B = 20 \cdot \log_{10} \frac{|d_{\max}|}{|d_{\min}|}, \tag{23.2}$$

where B is the dynamic range in decibels.

With the standard eight bits per colour channel, we have for a regular computer display $d_{\max} = 255$ and $d_{\min} = 1$, which leads to a dynamic range of $B' = 255$, or equivalently, $B \approx 48.1$ dB.

Below, when we speak of the dynamic range of a set of scalar data, we also refer to (23.2). In this case, we take $|d_{\max}|$ and $|d_{\min}|$ as the maximal and minimal nonzero data values (in terms of absolute value), respectively.

It is possible to extend the representable range by using a trick mentioned in, e.g., [35]. Because the colour channels are independent, one can define a colour palette utilizing the range of all three channels independently. However, this comes with the cost of nontrivial interpretation, and losing grayscale representability. Because grayscale print is still an important medium for scientific publishing, techniques which allow for easy grayscale conversion are preferable in the context of the present study.

Another classical engineering trick, which is not applicable to print media, is the use of pulse width modulation (PWM) to represent fractional pixel values [48]. The physical dynamic range of the display stays the same, but because fractional values are represented, less information is lost in quantization. The dynamic range extension comes from making the effective smallest representable pixel value $d_{\min}^{\text{eff}} \in (0, 1)$. When the refresh rate of the display device is high enough, the illusion can be convincing; otherwise the picture may flicker.

23.4 Tone Mapping Methods Used in the Present Study

In this study, we chose two tone reproduction operators from literature, and tested one of our own (specifically for HDR plotting). Minor changes (documented in this section) were made to the operators from literature, in order to adapt them into the HDR plotting context. For the full technical details, a GNU Octave compatible MATLAB implementation is available.[3]

All three methods are order-preserving; gradient reversal cannot occur. In all the methods, the data is first histogrammed, taking the logarithm and then binning linearly. This creates a logarithmic binning of the data. The methods then operate on the obtained (discrete) logarithmic density histogram.

Due to the logarithmic processing, positive-valued data, such as light intensity or probability density, is the easiest to handle. In practice, zeroes require some extra care. If the data to be displayed contains also negative values (such as PCM audio waveforms), there are two options. Taking the absolute value of the data, and handling it all at once, produces a scale that is symmetric with respect to the origin. Another option is to handle the positive and negative parts of the data separately, producing an independent scale for each. For the sake of simplicity, in this text we concentrate on positive data only.

All three methods are histogram remappers. The resulting remapped histogram is used the same way as in histogram equalization. For a quick review, let $p(x) : (0, +\infty) \mapsto [0, +\infty)$ be a density histogram (hence, piecewise constant) and $C(x) = \int_0^x p(\xi)\,\mathrm{d}\xi : (0, +\infty) \mapsto [0, 1]$ the corresponding cumulative histogram. The normalization is $\lim_{x\to+\infty} C(x) = 1$; in fact, this maximum will be reached at the end of the last histogram bin that contains a nonzero value.

[3] Files `hdr*.m` in https://yousource.it.jyu.fi/savu/codes/. Link cited 13 Jan 2012.

Given such a function $p(x)$, histogram equalization works by mapping $H(D) := C(D)$. In the mapped data, each element (pixel) belongs to the interval $[0, 1]$. This can then be scaled linearly to the range appropriate for the display device (and quantized to available pixel intensities). If the given function $p(x)$ is the actual density histogram of D, then the cumulative histogram of $H(D)$ will be a straight line from 0 to 1; the data will be histogram-equalized. The idea of the general histogram remapper is to first modify $p(x)$ in some appropriate way, before computing $H(D)$. See Figs. 23.4–23.5 below for an illustration of some mapping curves in a test example.

The first chosen method (below Method A) was based on [28]. The perceptual modelling was not included; only the contrast expansion limiting histogram remapping algorithm was used. To this, no modifications were needed. In this method, the growth in display intensity is limited from above to at most the growth in world intensity; i.e., contrast in the image is never expanded.[4] Contrast expansion is prevented by capping and renormalizing the histogram. This acts as a slope limiter for the cumulative histogram. By [28], the maximum allowed data value in one bin is

$$\text{ceiling} = T \frac{\Delta B}{B_{\text{display}}}, \tag{23.3}$$

where T is the current sum of the histogram data (values from all bins summed together), ΔB is the width of one histogram bin in decibels, and B_{display} is the dynamic range of the display device, as per (23.2). The quantity ΔB can be computed as $\Delta B = B_{\text{data}}/N$, where N is the number of histogram bins used and B_{data} is computed by (23.2). The bins are looped over, and any bin which has a higher data value than (23.3) is clipped to the ceiling. The excess is summed to a total removed. The whole histogram is then iterated over several times. Once the total removed during one iteration of the outer loop falls below a prescribed tolerance (in the present study, we chose 0.5 % of the largest value in the original, unmodified histogram data), the algorithm terminates. The updated histogram is then normalized so that it becomes a density histogram. For details, see the original paper.

The second method (Method B) was based on [18]. The original method is a local one, working directly on the 2D image data. A corresponding global method, applicable for HDR plotting, was created by simply applying the method in one dimension to the logarithmically binned histogram. This method is a gradient attenuator. First, a Gaussian pyramid is constructed, and finite differences are used at each level of the pyramid for gradient approximation. Then the gradient field at each level is updated adaptively, compressing large values of the gradient while (optionally) amplifying small ones. Finally, an auxiliary Poisson problem is solved in order to reconstruct scalar potentials that match the updated gradient fields as well as possible, and the solutions are assembled to form the output. The authors use a full multigrid (FMG) solver to obtain the solution in linear time. Our case is simpler;

[4] Similar ideas were explored ten years earlier in a medical imaging context by [38]; however, we use [28] since it explicitly provides an algorithm.

in one dimension, the Poisson problem can be skipped. The function corresponding to a given derivative field can obviously be found directly by numerical integration. The constant of integration and final scaling are then fixed by requiring that the resulting function is a density histogram.

The third method (Method C), for comparison, was based on a simple observation. The aim of the other two methods is to remove the highest peaks in the histogram, while retaining the overall shape approximately. Thus, it should be possible to retain the overall location of the most massive peaks, while smoothing out the histogram, by applying linear diffusion to the logarithmic histogram. The one-dimensional time-dependent heat equation was set up, with zero right-hand side, zero Neumann boundary conditions at both ends, and the original histogram as the initial condition. The diffusion simulation was run until the highest peak in the histogram (i.e. the L_∞ norm of the solution) fell under the ceiling (23.3).

It should be noted that Methods A and C will fail if $B_{\text{data}} \leq B_{\text{display}}$ (i.e. if the data is LDR); this is because (23.3) will then produce a ceiling that cannot be satisfied. This case must be detected at the outset before applying the methods.

For LDR data, Method C simplifies to logarithmic scaling. To see this, let $t \to +\infty$ in the linear diffusion simulation; the end result is a constant function in logarithmic histogram space. Thus, if $B_{\text{data}} \leq B_{\text{display}}$, we can skip the tone mapping and use logarithmic scaling instead. When this occurs, the display device has enough dynamic range to display the data without losing contrasts.

23.5 Results

The results from the tone mapping algorithms are shown in Fig. 23.1. After preclipping out anything smaller than 10^{-3}, the data has a dynamic range of $B \approx 94.8$ dB ($B' \approx 54750$). Recall that a display with 8-bit colour channels, without tone mapping, can display $B \approx 48.1$ dB ($B' = 255$). How the displayed HDR data was produced will be explained in the next section; for now, the item of interest is the relative performance of the different methods. The colour scale runs from white at zero to the data maximum at black. Note that the scale is neither linear nor logarithmic; as was discussed earlier, tone mapping methods produce a data-adaptive scaling.

We have also provided, as a baseline for comparison, corresponding plots using six naïve scalings in Figs. 23.2–23.3. These methods are, respectively, linear scaling (full range), linear scaling with 0.1 % of the top end (in terms of histogram mass) clipped out, logarithmic scaling (full range), logarithmic scaling showing the highest 48 dB only, the 10th root of the data,[5] and classical histogram equalization.

The updated histograms and corresponding cumulative histograms are provided in Figs. 23.4–23.5. The histograms are given for the histogram remapping methods only, plotted on a log-log scale (with zeroes deleted from data). However, since

[5] According to [18], root-taking is a popular naïve approach.

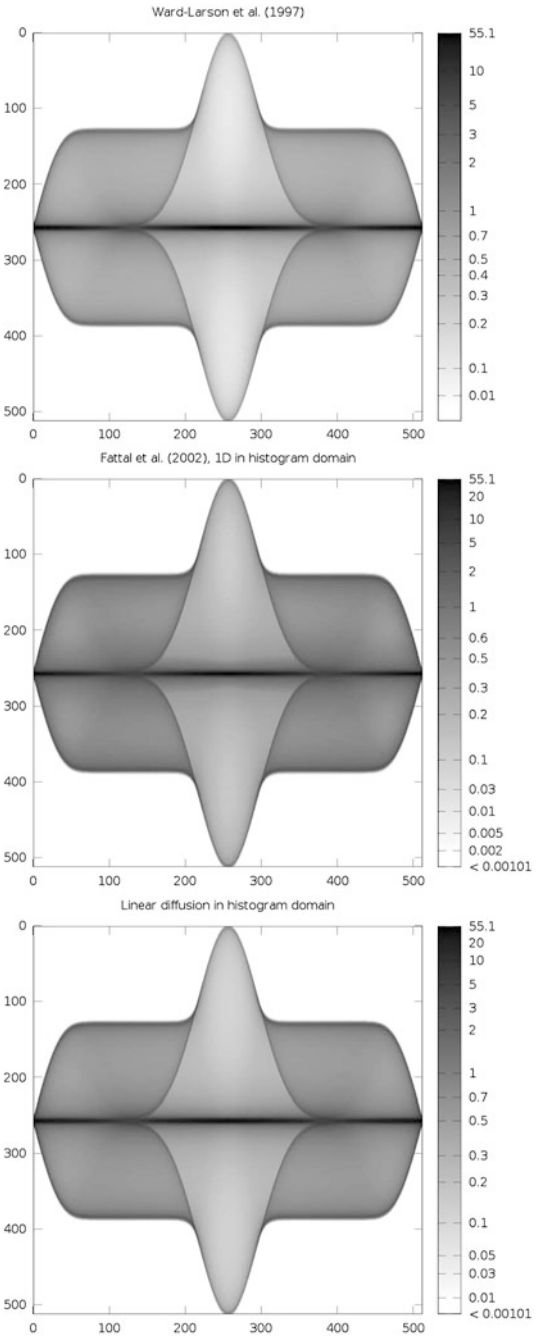


Fig. 23.1 HDR plotting, using tone mapping methods. Top to bottom: Method A [28], Method B [18], and Method C (linear diffusion). Note data-adaptive (non-linear, non-logarithmic) colour scale

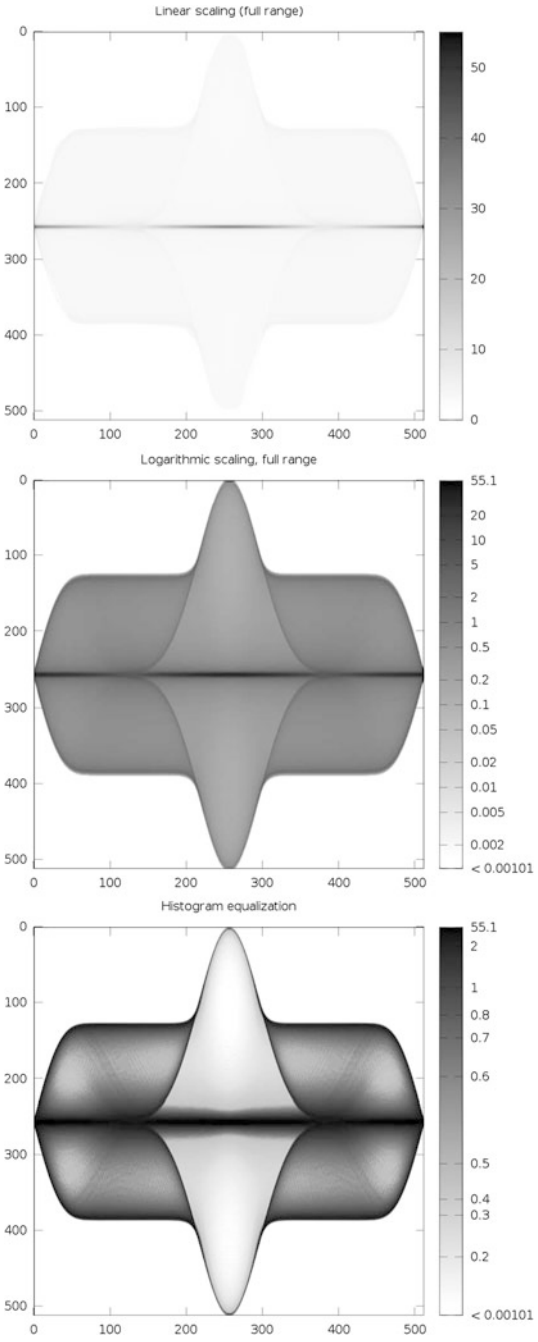


Fig. 23.2 HDR plotting, baseline (naïve) methods (set 1 of 2). Top to bottom: linear (*full scale*), logarithmic (*full scale*), histogram equalization

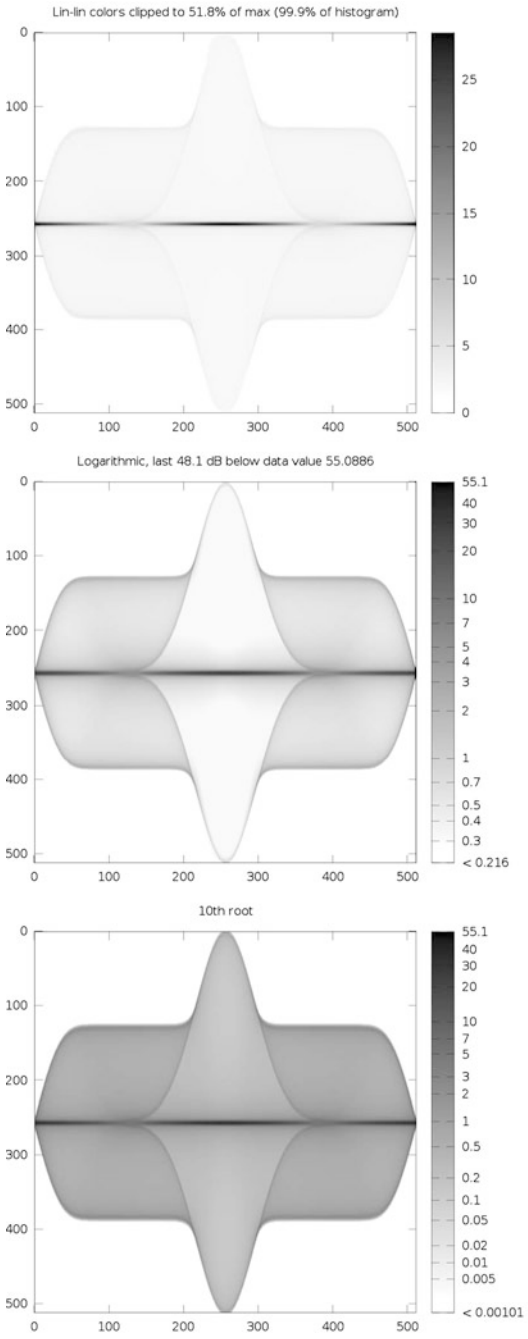


Fig. 23.3 HDR plotting, baseline (naïve) methods (set 2 of 2). Top to bottom: linear (showing only bottom 99.9 % of data), logarithmic (showing top 48 dB only), 10th root of data (showing all data)

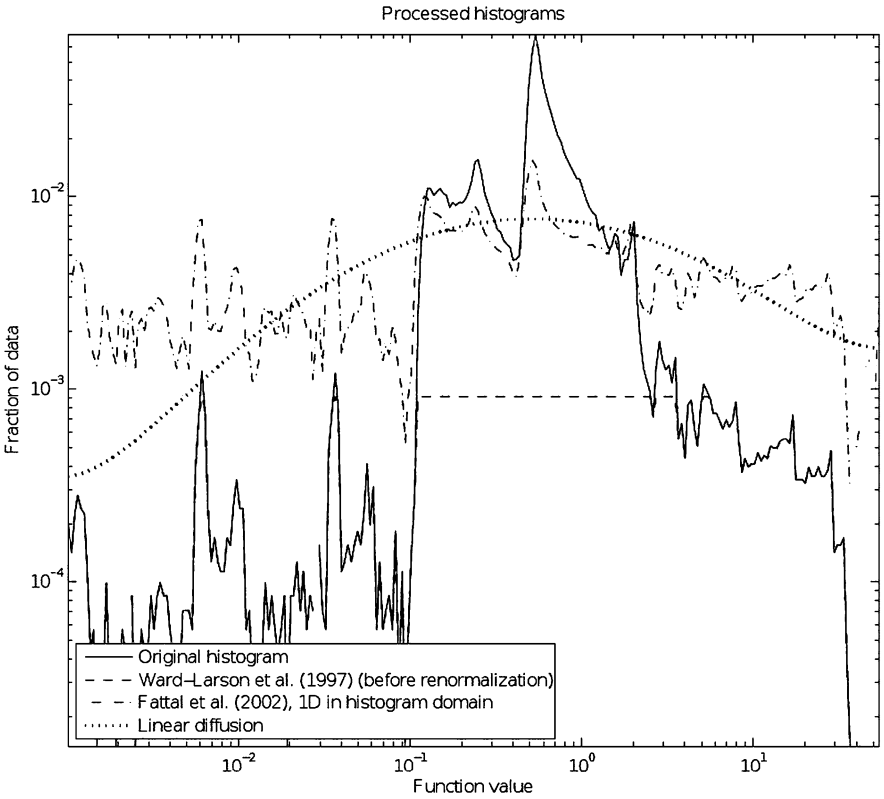


Fig. 23.4 Processed histograms of the methods used, corresponding to the images in Fig. 23.1. Note the log – log scale

the cumulative histogram is effectively the tone mapping function (as explained above), these are provided for the baseline methods, too. Here the scaling is log-lin; logarithmic in input data value (x axis) and linear in remapped output (y axis). Note that in the cumulative plot, the slope of the Ward-Larson mapping (Method A) never exceeds that of the logarithmic mapping of the last 48 dB; this is because the method prevents contrast expansion.

Method A gives excellent results; the detail in the HDR function can be seen very clearly. The results from Method B are similar, but consistently darker than those from Method A. It seems that Method B allocates a relatively larger portion of the available bins to the low end (small data values), leaving less bins for the high end (large data values). With the chosen colour scale, this makes the image darker. Also Method C produces acceptable results. Surprisingly, the differences in the results between the three different methods are relatively minor.

Since Method A gives, subjectively, slightly better results than the other two, and is also very simple to implement, our recommendation is to use Method A.

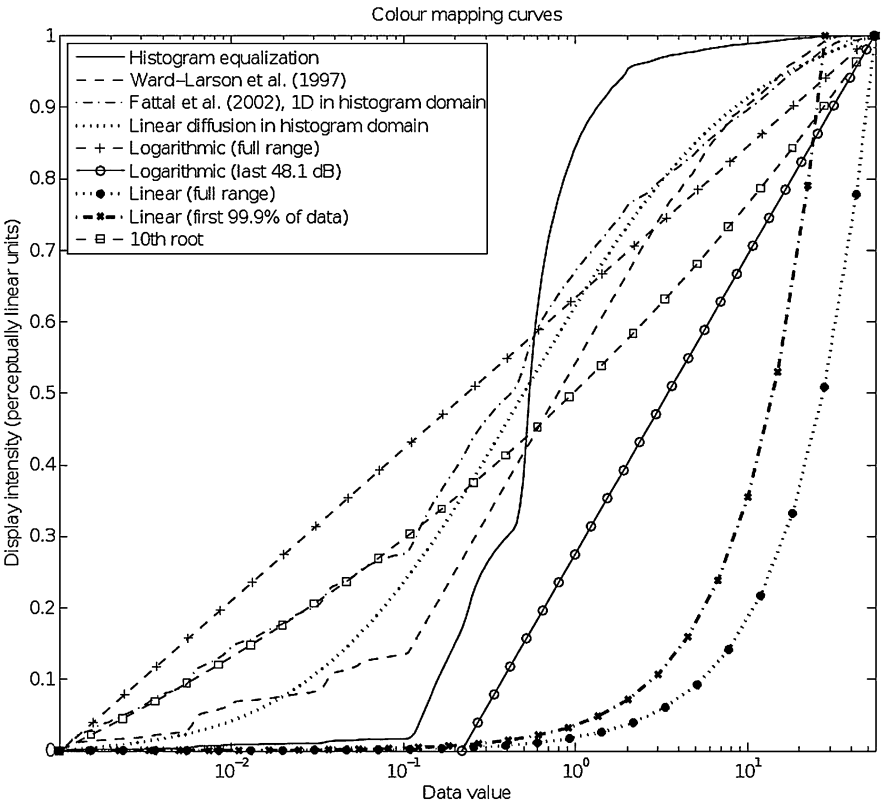


Fig. 23.5 Cumulative histograms (tone mapping functions) of the methods used, corresponding to the images in Figs. 23.1–23.3. Note the log–lin scale

It should be noted that all three methods can easily compress a dynamic range of over $B = 100$ dB ($B' = 10^5$) onto a regular display. This is not surprising, since the first two methods have been designed for use with HDR photography, where this is a requirement.

The methods were tested up to approximately $B = 300$ dB ($B' = 10^{15}$). While the end result in that case does not look as nice, it still gives some idea of the overall structure, see Fig. 23.6. (This is the same data as above, this time with the pre-clipping disabled. The extremely small numbers are ringing artifacts of the Fourier-based density estimator that produced the data.)

23.6 Application to Probability Densities in Mechanical Problems

As illustration of the use of HDR plotting in mechanics, we consider two vibration problems with uncertain data: transverse waves in a classical vibrating string, and

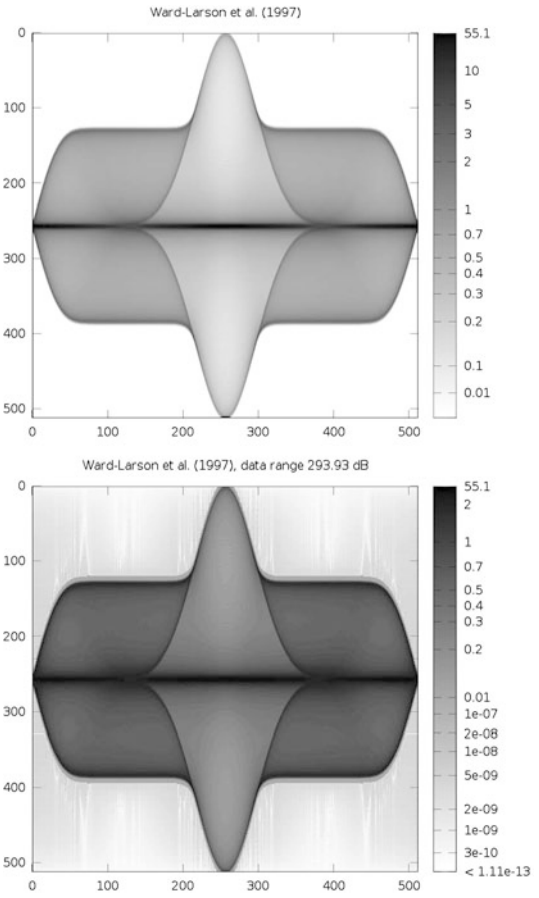


Fig. 23.6 A high dynamic range function tone mapped using Method A. *Top*: Pre-cutoff at 10^{-3}; dynamic range $B \approx 94.8$ dB. *Bottom*: No pre-cutoff; dynamic range $B \approx 294.0$ dB

the dynamic out-of-plane behaviour of an axially travelling panel submerged in axial potential flow.

For these problems, the behaviour of the displacement itself is of course not very interesting, because for a non-damped oscillator, any small disturbance in the frequency will cause two initially close trajectories to eventually diverge. Basically all that is needed for analysis are the eigenfrequencies, which behave in a stable manner. However, for the purposes of illustrating HDR plotting, we will consider displacements only, because this generates HDR data that is ideal for demonstration of the methods. From the modelling viewpoint, in the plotted time evolutions, we can visually observe the divergence of the solution set.

First, consider travelling transverse waves in a finite ideal string, with the endpoints fixed to zero level. As is well known, this situation is described by the one-dimensional wave equation $w_{tt} + cw_{xx} = 0$ for the displacement $w \equiv w(x,t)$. The boundary conditions are zero Dirichlet, $w(-1,t) = w(1,t) = 0$. As initial conditions, we choose an initial shape $w(x,0) \equiv w_0(x)$ (fulfilling the boundary conditions), and zero transverse velocity, $w_t(x,0) \equiv 0$. An analytical solution for this

case can be easily constructed by repeated reflection of the free-space d'Alembert solution; hence, we only need to plot a known function.

Now, let the wave propagation speed c be uncertain. We take $c = c_0 + X$, where X is a random variable with a zero-mean Gaussian distribution truncated at $\pm 3\sigma$. The parameter $c_0 = 0.05$ and σ is taken as 1 % of c_0. We then use SAVU to compute and plot the resulting probability density. See Figs. 23.7–23.8 for a time evolution simulation. (The last frame of this simulation is the example used in the previous section.)

The second problem comes from our research. We consider an axially travelling panel submerged in ideal fluid (potential flow), with an optional axial free-stream component. Details can be found in [3, 4, 26]. *Panel* is understood as a plate in the limit of cylindrical deformation (the *flat panel* of aeroelasticity; see, e.g., [6]).

We take as the uncertain parameters the axial panel velocity V_0, applied axial tension T, the Young modulus of the panel E, and the mass per unit area of the panel m. A Gaussian distribution truncated at $\pm 3\sigma$ is used for each parameter, representing a typical measurement error. The input is thus a four-dimensional hypercube. We choose σ (arbitrarily) as 1 % of the reference value for each parameter. The governing equation and reference values for the parameters can be found in [26, p. 100 and 155]. The initial conditions for w and w_t are taken as zero, and an external disturbance (force) is applied for a finite time at the beginning.

In Fig. 23.9 we have snapshots of the time behaviour of the displacement, with the four simultaneously uncertain parameters.

23.7 Conclusion

In this study, we presented and tested three methods for data-adaptive dynamic range compression of high dynamic range (HDR) mathematical functions, achieving a visual contrast preserving representation on low dynamic range media such as regular computer displays and print. This produced a scaling that is neither linear nor logarithmic, but instead data-adaptive. It is also global across the picture, allowing a colour bar to be created in the usual manner. One of the methods was seen to perform slightly better than the other two. As it was also the simplest to implement, it was recommended.

High dynamic range functions occur, for example, as probability densities in some uncertain data problems in mechanics. When such data is encountered, in our opinion it is natural to look for a visualization that shows the structure clearly. Compared to logarithmic scaling, the described methods have the advantage of visual contrast preservation, making the structure of the function more clearly visible.

Finally, it should be especially emphasized that what the presented methods do is dynamic range compression, and dynamic range compression only. For data that already fits into the dynamic range of the display (LDR data), classical methods are sufficient. In that case, these methods will either do nothing or, in the worst case, possibly harm the visualization. But if the data to be visualized is HDR, then these methods are very useful and can significantly improve the visualization.

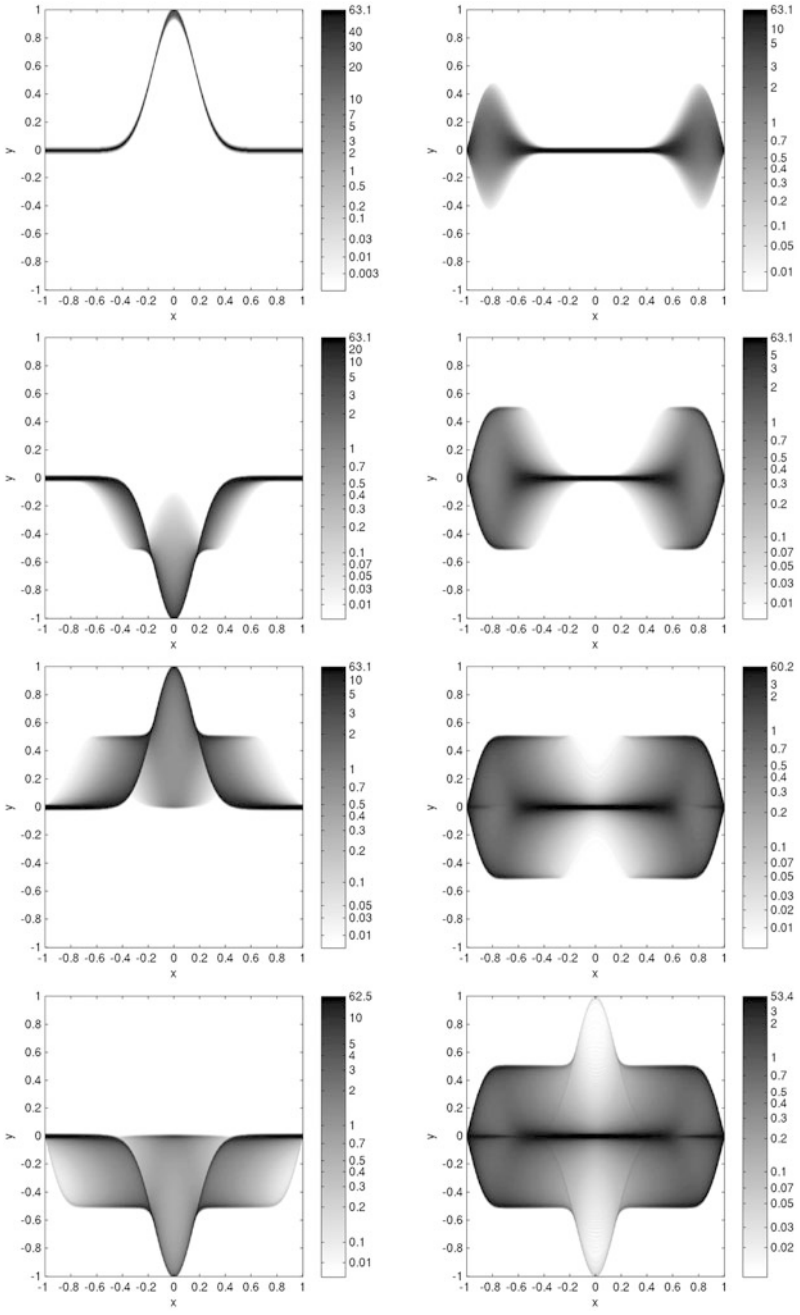


Fig. 23.7 Probability density of travelling transverse waves in a string with uncertain wave propagation speed. Snapshots taken at regular intervals; time increases as top left, top right, middle left and so on. *Top left*: initial pulse

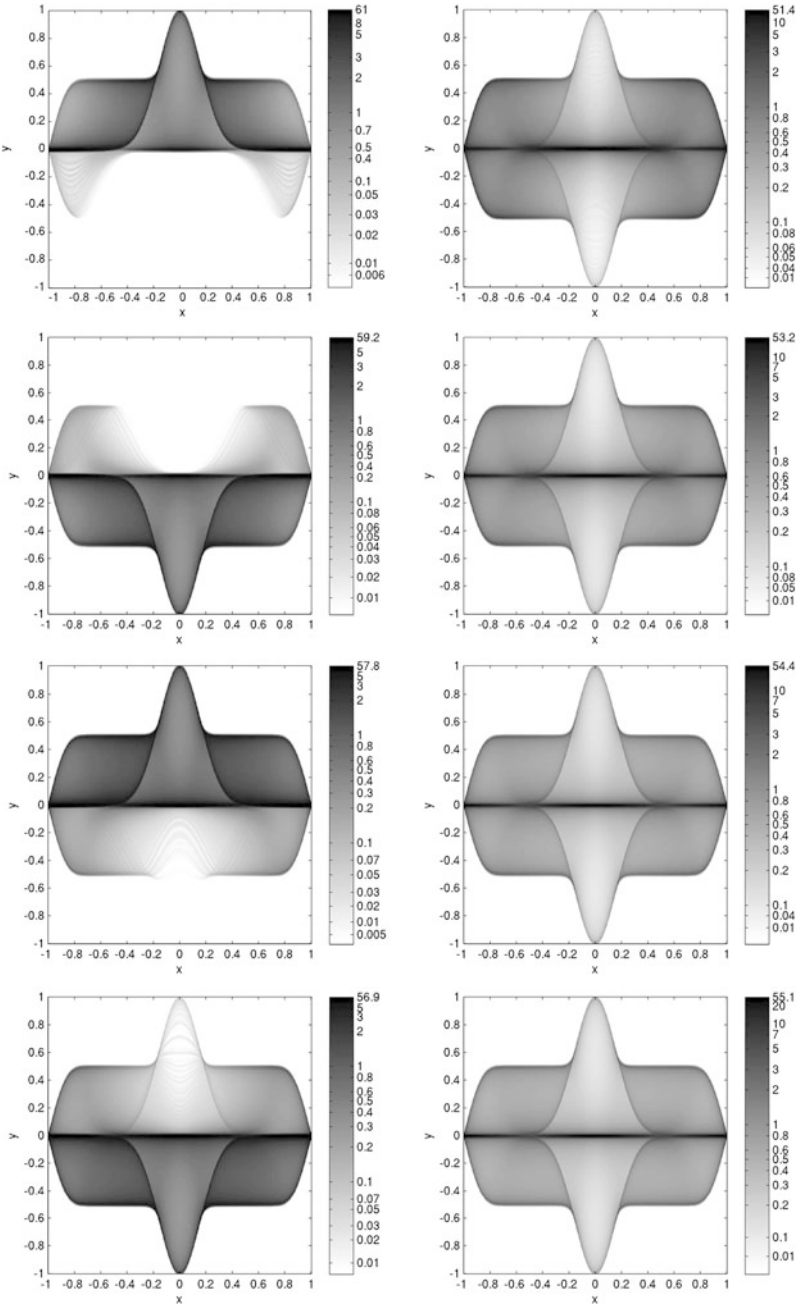


Fig. 23.8 Probability density of travelling transverse waves in a string with uncertain wave propagation speed. Snapshots taken at regular intervals; continued from Fig. 23.7

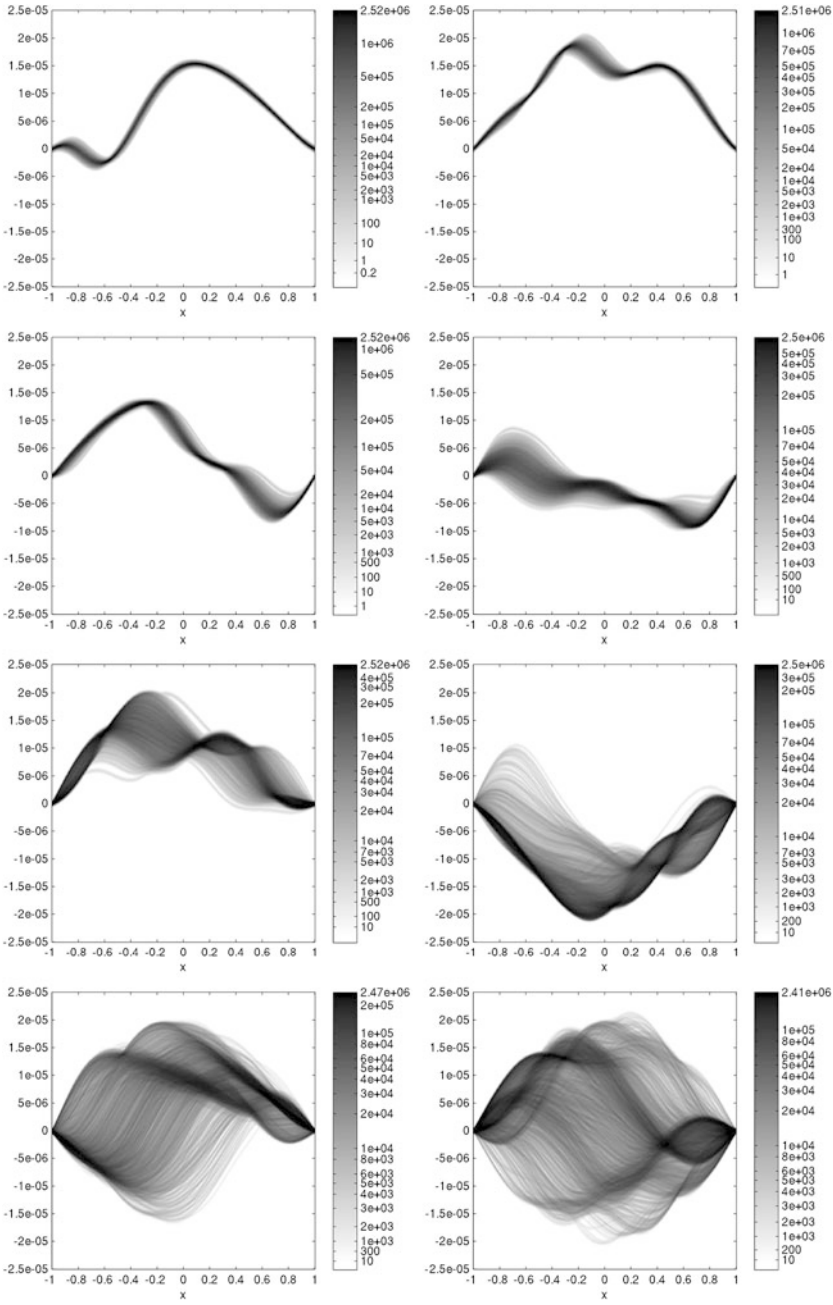


Fig. 23.9 Probability density of transverse displacement of an axially travelling panel submerged in ideal fluid, with four simultaneously uncertain parameters. After cutting the low end at 10^{-1}, the dynamic range is approximately $B \approx 147.6$ dB ($B' \approx 2.4 \cdot 10^7$). Snapshots taken at different times. Left to right by row: $t = 5,10$; 15,20; 50,100; 150 and 200 s

Hence, the presented methods are not intended to replace classical scalings, but to provide an extension where needed. By utilizing dynamic range compression, it is possible to extend the range of mathematical functions that can be represented on regular computer displays and print in a visual contrast preserving manner.

References

1. Abanoz B, Wang M (2008) A review of high dynamic range imaging on static scenes. Technical report 2008-04, Boston University
2. Ashikhmin M (2002) A tone mapping algorithm for high contrast images. In: Proceedings of the 13th eurographics workshop on rendering, pp 145–156. The Eurographics Association
3. Banichuk N, Jeronen J, Neittaanmäki P, Tuovinen T (2010) Static instability analysis for travelling membranes and plates interacting with axially moving ideal fluid. J Fluids Struct 26(2):274–291. doi:10.1016/j.jfluidstructs.2009.09.006
4. Banichuk N, Jeronen J, Neittaanmäki P, Tuovinen T (2011) Dynamic behaviour of an axially moving plate undergoing small cylindrical deformation submerged in axially flowing ideal fluid. J Fluids Struct 27(7):986–1005. doi:10.1016/j.jfluidstructs.2011.07.004
5. Beachkofski BK, Grandhi R (2002) Improved distributed hypercube sampling. In: Proceedings of the 43rd conference AIAA/ASME/ASCE/AHS/ASC on structures, dynamics and materials. Paper AIAA-2002-1274
6. Bisplinghoff RL, Ashley H (1975) Principles of aeroelasticity, 2nd edn. Dover, New York
7. Botev ZI, Grotowski JF, Kroese DP (2010) Kernel density estimation via diffusion. Ann Stat 38(5):2916–2957
8. Chacón JE, Duong T (2010) Multivariate plug-in bandwidth selection with unconstrained pilot bandwidth matrices. Test 19(2):375–398
9. Cox SE, Booth DT (2009) Shadow attenuation with high dynamic range images. Environ Monit Assess 158(1–4):231–241
10. Debevec P (1998) Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: Proceedings of the 25th annual conference on computer graphics and interactive techniques (SIGGRAPH'98). ACM, New York, pp 189–198
11. Debevec P, Malik J (1997) Recovering high dynamic range radiance maps from photographs. In: ACM SIGGRAPH 2008 classes (SIGGRAPH'08). ACM, New York. Article No. 31,
12. Devlin K (2002) A review of tone reproduction techniques. Technical report CSTR-02-005, University of Bristol
13. Drago F, Martens W, Myszkowski K, Seidel H-P (2002) Perceptual evaluation of tone mapping operators with regard to similarity and preference. Research report MPI-I-2002-4-002, Max-Planck-Institut für Informatik
14. Drago F, Myszkowski K, Annen T, Chiba N (2003) Adaptive logarithmic mapping for displaying high contrast scenes. Comput Graph Forum 22(3):419–426. doi:10.1111/1467-8659.00689
15. Duan J, Bressan M, Dance C, Qiu G (2010) Tone-mapping high dynamic range images by novel histogram adjustment. Pattern Recognit 43(5):1847–1862. doi:10.1016/j.patcog.2009.12.006
16. Duong T, Hazelton ML (2003) Plug-in bandwidth matrices for bivariate kernel density estimation. J Nonparametr Stat 15(1):17–30
17. Durand F, Dorsey J (2002) Fast bilateral filtering for the display of high-dynamic-range images. In: Proceedings of the 29th annual conference on computer graphics and interactive techniques (SIGGRAPH'02). ACM, New York, pp 257–266

18. Fattal R, Lischinski D, Werman M (2002) Gradient domain high dynamic range compression. In: Proceedings of the 29th annual conference on computer graphics and interactive techniques (SIGGRAPH'02). ACM, New York, pp 249–256
19. Finlayson G, Hordley S, Schaefer G, Tian GY (2005) Illuminant and device invariant colour using histogram equalisation. Pattern Recognit 38(2):179–190. doi:10.1016/j.patcog.2004.04.010
20. Gatta C, Rizzi A, Marini D (2007) Perceptually inspired HDR images tone mapping with color correction. Int J Imaging Syst Technol 17(5):285–294
21. Goodnight N, Wang R, Woolley C, Humphreys G (2003) Interactive time-dependent tone mapping using programmable graphics hardware. In: Proceedings of the 14th eurographics symposium on rendering (EGSR'03), pp 26–37 Eurographics Association
22. Goshtasby AA High dynamic range reduction via maximization of image information, 2003. http://www.cs.wright.edu/people/faculty/agoshtas/goshtasby_hdr.pdf. CiteSeerX: doi:10.1.1.121.421
23. Helton JC, Davis FJ (2002) Illustration of sampling-based methods for uncertainty and sensitivity analysis. Risk Anal 22(3):591–622
24. Helton JC, Davis FJ (2002) Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. Sandia report SAND2001-0417, Sandia National Laboratories
25. Helton JC, Johnson JD, Sallaberry CJ, Storlie CB (2006) Survey of sampling-based methods for uncertainty and sensitivity analysis. Sandia report SAND2006-2901, Sandia National Laboratories
26. Jeronen J (2011) On the mechanical stability and out-of-plane dynamics of a travelling panel submerged in axially flowing ideal fluid: a study into paper production in mathematical terms. PhD thesis, University of Jyväskylä, Jyväskylä. http://julkaisut.jyu.fi/?id=978-951-39-4596-1
27. Krawczyk G, Myszkowski K, Seidel H-P (2005) Perceptual effects in real-time tone mapping. In: Proceedings of the 21st spring conference on computer graphics (SCCG'05). ACM, New York, pp 195–202
28. Larson GW, Rushmeier H, Piatko C (1997) Visibility matching tone reproduction operator for high dynamic range scenes. IEEE Trans Vis Comput Graph 3(4):291–306
29. Mann S, Picard RW (1995) On being 'undigital' with digital cameras: extending dynamic range by combining differently exposed pictures. In: Proceedings the 48th annual conference of IS&T, pp 442–448
30. Mantiuk R, Daly S, Kerofsky L (2008) Display adaptive tone mapping. ACM Trans Graph (TOG) – Proc ACM SIGGRAPH 2008, 27(3). doi:10.1145/1360612.1360667
31. Mantiuk R, Seidel H-P (2008) Modeling a generic tone-mapping operator. Comput Graph Forum 27(2):699–708. doi:10.1111/j.1467-8659.2008.01168.x
32. McCann JJ, Rizzi A (2007) Veiling glare: the dynamic range limit of HDR images. In: Human vision and electronic imaging XII. Proceedings of electronic imaging science and technology, vol 6492. IS&T and SPIE
33. McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21(2):239–245
34. Owen Art B (1992) Orthogonal arrays for computer experiments, integration and visualization. Stat Sin 2(2):439–452
35. Park SH, Montag ED (2007) Evaluating tone mapping algorithms for rendering non-pictorial (scientific) high-dynamic-range images. J Vis Commun Image Represent 18(5):415–428
36. Parks DR, Roederer M, Moore WA (2006) A new "logicle" display method avoids deceptive effects of logarithmic scaling for low signals and compensated data. Cytometry A 69(6):541–551
37. Pattanaik SN, Tumblin J, Yee H, Greenberg DP (2000) Time-dependent visual adaptation for fast realistic image display. In: Proceedings of the 27th annual conference on computer graphics and interactive techniques (SIGGRAPH'00). ACM, New York, pp 47–54
38. Pizer SM, Amburn EP, Austin JD, Cromartie R, Geselowitz A, Greer T, Romeny BTH, Zimmerman JB, Zuiderveld K (1987) Adaptive histogram equalization and its variations. Comput

Vis Graph Image Process 39(3):355–368
39. Reinhard E, Devlin K (2005) Dynamic range reduction inspired by photoreceptor physiology. IEEE Trans Vis Comput Graph 11(1):13–24
40. Reinhard E, Kunkel T, Marion Y, Brouillat J, Cozot R, Bouatouch K (2007) Image display algorithms for high and low dynamic range display devices. J Soc Inf Disp 15(12):997–1014
41. Schlick C (1994) Quantization techniques for visualization of high dynamic range pictures. In: Photorealistic rendering techniques. Proceedings of the 5th eurographics workshop on rendering. Springer, Berlin, pp 7–20
42. Shan Q, Jia J, Brown MS (2010) Globally optimized linear windowed tone mapping. IEEE Trans Vis Comput Graph 16(4):663–675
43. Sheather SJ, Jones MC (1991) A reliable data-based bandwidth selection method for kernel density estimation. J R Stat Soc B 53(3):683–690
44. Smith K, Krawczyk G, Myszkowski K, Seidel H-P (2006) Beyond tone mapping: enhanced depiction of tone mapped HDR images. Comput Graph Forum 25(3):427–438
45. Talvala E-V, Adams A, Horowitz M, Levoy M (2007) Veiling glare in high dynamic range imaging. ACM Trans Graph (TOG) – Proc ACM SIGGRAPH 2008, 26(3)
46. Tang B (1993) Orthogonal array-based Latin hypercubes. J Am Stat Assoc 88(424):1392–1397
47. Tumblin J, Hodgins JK, Guenter BK (1999) Two methods for display of high contrast images. ACM Trans Graph 18(1):56–94
48. Wu Y, Qiu B (2010) Perceptually fractural pixel values in rendering high dynamic range images. In: Proc SPIE, vol 7744. doi:10.1117/12.863019
49. Yoshida A, Blanz V, Myszkowski K, Seidel H-P (2005) Perceptual evaluation of tone mapping operators with real-world scenes. In: Human vision and electronic imaging X. Proceedings of the SPIE, vol 5666. SPIE, New York, pp 192–203
50. Yoshida A, Mantiuk R, Myszkowski K, Seidel H-P (2006) Analysis of reproducing real-world appearance on displays of varying dynamic range. Comput Graph Forum 25(3):415–426
51. Yuan X, Nguyen MX, Chen B, Porter DH (2005) High dynamic range volume visualization. In: Proceedings of the conference on visualization 2005. IEEE Comput Sci, Los Alamitos, pp 327–334

Chapter 24
Failure Simulations with a Strain Rate Dependent Ductile-to-Brittle Transition Model

Juha Hartikainen, Kari Kolari, and Reijo Kouhia

Abstract In this paper, simulations with a phenomenological model to describe the ductile-to-brittle transition of rate-dependent solids are presented. The model is based on consistent thermodynamic formulation using proper expressions for the Helmholtz free energy and the dissipation potential. In the model, the dissipation potential is additively split into damage and visco-plastic parts and the transition behaviour is obtained using a stress dependent damage potential. The damage is described by using a vectorial variable.

Keywords Constitutive model · Continuum damage mechanics · Viscoplasticity · Dissipation potential · Ductile-to-brittle transition

24.1 Introduction

Most materials exhibit rate-dependent inelastic behaviour. An increasing strain rate usually increases the yield stress thus enlarging the elastic range. However, the ductility is gradually lost and for some materials there exists a rather sharp transition strain rate zone after which the material behaviour is completely brittle.

In this paper, a phenomenological approach to model the ductile-to-brittle transition of rate-dependent solids is presented. It is an extension to the model presented in [1, 5] using a vectorial damage variable [8]. The model is based on consistent thermodynamic formulation using proper expressions for the Helmholtz free energy and dissipation potential. The dissipation potential is additively split into damage

J. Hartikainen (✉)
Aalto University, P.O. Box 12100, 00076 Aalto, Finland
e-mail: juha.hartikainen@aalto.fi

K. Kolari
VTT, P.O. Box 1000, 02044 VTT, Finland
e-mail: kari.kolari@vtt.fi

R. Kouhia
Tampere University of Technology, P.O. Box 589, 33101 Tampere, Finland
e-mail: reijo.kouhia@tut.fi

S. Repin et al. (eds.), *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, Computational Methods in Applied Sciences 27,
DOI 10.1007/978-94-007-5288-7_24,

and visco-plastic parts and the transition behaviour is obtained using a stress dependent damage potential. The basic features of the model are discussed.

24.2 Thermodynamic Formulation

The constitutive model is derived using a thermodynamic formulation, in which the material behaviour is described completely through the Helmholz free energy and the dissipation potential in terms of the variables of state and dissipation and considering that the Clausius-Duhem inequality is satisfied [6].

The Helmholtz free energy

$$\psi = \psi(\boldsymbol{\varepsilon}_\mathrm{e}, \mathbf{D})$$

is assumed to be a function of the elastic strains, $\boldsymbol{\varepsilon}_\mathrm{e}$, and the damage vector $\mathbf{D}$. Assuming small strains, the total strain can be additively decomposed into elastic and inelastic strains $\boldsymbol{\varepsilon}_\mathrm{i}$ as $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_\mathrm{e} + \boldsymbol{\varepsilon}_\mathrm{i}$.

The Clausius-Duhem inequality, in the absence of thermal effects, is formulated as

$$\gamma \geq 0, \qquad \gamma = -\rho\dot{\psi} + \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}, \tag{24.1}$$

where ρ is the material density. As usual in the solid mechanics, the dissipation potential

$$\varphi = \varphi(\boldsymbol{\sigma}, \mathbf{Y})$$

is expressed in terms of the thermodynamic forces $\boldsymbol{\sigma}$ and $\mathbf{Y}$ dual to the fluxes $\dot{\boldsymbol{\varepsilon}}_\mathrm{i}$ and $\dot{\mathbf{D}}$, respectively. The dissipation potential is associated with the power of dissipation, γ, such that

$$\gamma = \frac{\partial \varphi}{\partial \boldsymbol{\sigma}} : \boldsymbol{\sigma} + \frac{\partial \varphi}{\partial \mathbf{Y}} \cdot \mathbf{Y}. \tag{24.2}$$

Using the definition (24.2), Eq. $(24.1)_2$, and defining that $\rho \partial \psi / \partial \mathbf{D} = -\mathbf{Y}$, results in the equation

$$\left(\boldsymbol{\sigma} - \rho \frac{\partial \psi}{\partial \boldsymbol{\varepsilon}_\mathrm{e}}\right) : \dot{\boldsymbol{\varepsilon}}_\mathrm{e} + \left(\dot{\boldsymbol{\varepsilon}}_\mathrm{i} - \frac{\partial \varphi}{\partial \boldsymbol{\sigma}}\right) : \boldsymbol{\sigma} + \left(\dot{\mathbf{D}} - \frac{\partial \varphi}{\partial \mathbf{Y}}\right) \cdot \mathbf{Y} = 0. \tag{24.3}$$

Then, if (24.3) holds for any evolution of $\dot{\varepsilon}_\mathrm{e}$, σ and Y, the inequality (24.1) is satisfied and the following relevant constitutive relations are obtained:

$$\boldsymbol{\sigma} = \rho \frac{\partial \psi}{\partial \boldsymbol{\varepsilon}_\mathrm{e}}, \qquad \dot{\boldsymbol{\varepsilon}}_\mathrm{i} = \frac{\partial \varphi}{\partial \boldsymbol{\sigma}}, \qquad \dot{\mathbf{D}} = \frac{\partial \varphi}{\partial \mathbf{Y}}. \tag{24.4}$$

24.3 Particular Model

In the present formulation, the Helmholtz free energy, ψ, is a function depending on the symmetric second-order strain tensor $\boldsymbol{\varepsilon}_\mathrm{e}$ and the damage vector $\mathbf{D}$. The integrity basis thus consists of the following six invariants:

$$\begin{aligned} &I_1 = \operatorname{tr}\boldsymbol{\varepsilon}_\mathrm{e}, \qquad I_2 = \frac{1}{2}\operatorname{tr}\boldsymbol{\varepsilon}_\mathrm{e}^2, \qquad I_3 = \frac{1}{3}\operatorname{tr}\boldsymbol{\varepsilon}_\mathrm{e}^3, \qquad I_4 = \|\mathbf{D}\|, \\ &I_5 = \mathbf{D}\cdot\boldsymbol{\varepsilon}_\mathrm{e}\cdot\mathbf{D}, \qquad I_6 = \mathbf{D}\cdot\boldsymbol{\varepsilon}_\mathrm{e}^2\cdot\mathbf{D}. \end{aligned} \tag{24.5}$$

A particular expression for the free energy, describing the elastic material behaviour with the directional reduction effect due to damage, is given by [8]

$$\begin{aligned} \rho\psi = (1-I_4)\left(\frac{1}{2}\lambda I_1^2 + 2\mu I_2\right) + H\big(\sigma^\perp\big)\frac{\lambda\mu}{\lambda+2\mu}\big(I_4 I_1^2 - 2I_1 I_5 I_4^{-1} + I_5^2 I_4^{-3}\big) \\ + \big(1-H\big(\sigma^\perp\big)\big)\left(\frac{1}{2}\lambda I_4 I_1^2 + \mu I_5^2 I_4^{-3}\right) + \mu\big(2I_4 I_2 + I_5^2 I_4^{-3} - 2I_6 I_4^{-1}\big), \end{aligned} \tag{24.6}$$

where λ and μ are the Lamé parameters, H is the Heaviside step function and

$$\sigma^\perp = \lambda I_1 + 2\mu\hat{\mathbf{D}}\cdot\boldsymbol{\varepsilon}_\mathrm{e}\cdot\hat{\mathbf{D}}, \quad \text{and} \quad \hat{\mathbf{D}} = \mathbf{D}/I_4. \tag{24.7}$$

To model the ductile-to-brittle transition due to an increasing strain rate, the dissipation potential is decomposed into the brittle damage part, φ_d, and the ductile viscoplastic part, φ_vp, as

$$\varphi(\boldsymbol{\sigma}, \mathbf{Y}) = \varphi_\mathrm{d}(\mathbf{Y})\varphi_\mathrm{tr}(\boldsymbol{\sigma}) + \varphi_\mathrm{vp}(\boldsymbol{\sigma}), \tag{24.8}$$

where the transition function, φ_tr, deals with the change in the mode of deformation when the strain rate $\dot{\boldsymbol{\varepsilon}}_\mathrm{i}$ increases. Applying an overstress type of viscoplasticity [2, 13, 14] and the principle of strain equivalence [11, 12], the following choices are made to characterize the inelastic material behaviour:

$$\varphi_\mathrm{d} = \frac{1}{2r+2}\frac{Y_\mathrm{r}^2}{\tau_\mathrm{d}(1-I_4)}H(\varepsilon_1 - \varepsilon_\mathrm{tresh})\left(\frac{(\mathbf{Y}-\mathbf{Y}_0)\cdot\mathbf{M}\cdot(\mathbf{Y}-\mathbf{Y}_0)}{Y_\mathrm{r}^2}\right)^{r+1}, \tag{24.9}$$

$$\varphi_\mathrm{tr} = \frac{1-I_4}{pn}\left[\frac{1}{\tau_\mathrm{vp}\eta}\left(\frac{\bar{\sigma}}{(1-I_4)\sigma_\mathrm{r}}\right)^p\right]^n, \tag{24.10}$$

$$\varphi_\mathrm{vp} = \frac{1}{p+1}\frac{(1-I_4)\sigma_\mathrm{r}}{\tau_\mathrm{vp}}\left(\frac{\bar{\sigma}}{(1-I_4)\sigma_\mathrm{r}}\right)^{p+1}, \tag{24.11}$$

where the parameters τ_d, r and n are associated with the damage evolution, and the parameters τ_vp and p with the visco-plastic flow. In addition, η denotes the inelastic transition strain rate and $\mathbf{Y}_0 = \beta Y_\mathrm{r}\mathbf{n}$, where β is a small number, acts as a seed for the

damage evolution, and $\mathbf{n}$ is the eigenvector of the elastic strain tensor corresponding to the largest principal strain ε_1. The damage threshold strain is $\varepsilon_{\text{tresh}}$. The direction of the damage vector is defined through the tensor $\mathbf{M} = \mathbf{n} \otimes \mathbf{n}$ where $\otimes$ denotes the tensor product. The relaxation times τ_{d} and τ_{vp} have the dimension of time and the exponents r, $p \geq 0$ and $n \geq 1$ are dimensionless. $\bar{\sigma}$ is a scalar function of stress, e.g. the effective stress $\sigma_{\text{eff}} = \sqrt{3J_2}$, where J_2 is the second invariant of the deviatoric stress. The reference values Y_{r} and σ_{r} can be chosen arbitrarily, and they are used to make the expressions dimensionally reasonable. Since only isotropic elasticity is considered, the reference value Y_{r} has been chosen as

$$Y_{\text{r}} = \sigma_{\text{r}}^2 / E, \tag{24.12}$$

where E is the Young modulus.

Making use of (24.4), the choices (24.6)–(24.11) yield the desired constitutive equations.

24.4 On the Integration Algorithms

There are many different algorithms for the integration of inelastic constitutive models. However, the fully implicit backward Euler scheme seems to be the most popular, although it is only first-order accurate [15–17]. In practical problems, especially in those of creep and viscoplasticity, the time steps are often large, several magnitudes larger than the critical time step of some explicit methods, e.g. the forward Euler method. Therefore, the integrator should be unconditionally stable and sufficiently accurate for large time steps.

As shown in [10], the asymptotic convergence rate does not necessarily reflect high accuracy outside the asymptotic range, which usually means step sizes smaller than the critical time step of the explicit Euler method. For large time steps, the first-order accurate backward Euler method seems to be more accurate than many higher-order schemes. Therefore, an integrator for inelastic constitutive models should be at least [9, 10]:

- L-stable
- and for $\dot{\sigma} + \lambda\sigma = 0$, $\lambda =$ constant, the stability function should be
  - strictly positive, and
  - monotonous with respect to time step.

It is obvious that the standard backward Euler scheme fulfils these requirements.

When damage is included in the constitutive model, behaviour of the solution of the governing evolution equations is completely different from that of viscous and plastic solutions. Solutions of problems in creep, plasticity, and viscoplasticity are diffusive and decay exponentially with time whereas damage produces a reactive type of solutions growing exponentially with time [3].

24.4.1 Standard Backward Euler Scheme

For rate-dependent solids implicit time integrators are preferable. In this study, the backward Euler scheme is used to integrate the constitutive model at the integration point level. Although the backward Euler scheme is asymptotically only first-order accurate, it has good accuracy properties for large, practically relevant time steps [10].

Using matrix notation, the constitutive model (24.4) is rewritten in the form

$$\dot{\boldsymbol{\sigma}} = \mathbf{f}_\sigma(\boldsymbol{\sigma}, \mathbf{D}), \tag{24.13}$$

$$\dot{\mathbf{D}} = \mathbf{f}_D(\boldsymbol{\sigma}, \mathbf{D}) \tag{24.14}$$

such that

$$\mathbf{f}_\sigma(\boldsymbol{\sigma}, \mathbf{D}) = \mathbf{C}(\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}_\mathrm{i}) + \frac{\partial \mathbf{C}}{\partial \mathbf{D}} \mathbf{C}^{-1} \boldsymbol{\sigma}, \tag{24.15}$$

$$\mathbf{f}_D(\boldsymbol{\sigma}, \mathbf{D}) = -\frac{\varphi_\mathrm{tr} H(\varepsilon_1 - \varepsilon_\mathrm{tresh})}{\tau_\mathrm{d}(1 - I_4)} \left(\frac{(\mathbf{Y} - \mathbf{Y}_0) \cdot \mathbf{M} \cdot (\mathbf{Y} - \mathbf{Y}_0)}{Y_\mathrm{r}^2} \right)^r, \tag{24.16}$$

where the elastic stress is $\boldsymbol{\sigma}_\mathrm{e} = \mathbf{C} : \boldsymbol{\varepsilon}_\mathrm{e}$, and the elastic constitutive matrix $\mathbf{C}$ of a damaged solid can be most conveniently written using the tensor component representation

$$\begin{aligned} C_{ijkl} = {} & \lambda\big(1 - \tilde{\lambda} I_4 H\big(\sigma^\perp\big)\big)\delta_{ij}\delta_{kl} + 2\mu\big[\delta_{ik}\delta_{jl} - \tilde{\lambda} I_4 H\big(\sigma^\perp\big)(\delta_{ij}\hat{D}_i\hat{D}_j + \hat{D}_i\hat{D}_j\delta_{kl})\big] \\ & + 2\mu\big[2 + (\tilde{\lambda} - 1)H\big(\sigma^\perp\big)\big] I_4 \hat{D}_i \hat{D}_j \hat{D}_k \hat{D}_l \\ & - 2\mu I_4(\delta_{il}\hat{D}_j\hat{D}_k + \delta_{jk}\hat{D}_i\hat{D}_l), \end{aligned} \tag{24.17}$$

where $\tilde{\lambda} = \lambda/(\lambda + 2\mu)$.

Applying the backward Euler scheme and the Newton linearisation method to the evolution equations (24.13) and (24.14) results in the linear system of equations[1]

$$\begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \begin{Bmatrix} \delta\boldsymbol{\sigma} \\ \delta\mathbf{D} \end{Bmatrix} = \Delta t \begin{Bmatrix} \mathbf{f}_\sigma \\ \mathbf{f}_D \end{Bmatrix} - \begin{Bmatrix} \Delta\boldsymbol{\sigma} \\ \Delta\mathbf{D} \end{Bmatrix}, \tag{24.18}$$

where

$$\mathbf{H}_{11} = \mathbf{I} - \Delta t \frac{\partial \mathbf{f}_\sigma}{\partial \boldsymbol{\sigma}}, \qquad \mathbf{H}_{12} = -\Delta t \frac{\partial \mathbf{f}_\sigma}{\partial \mathbf{D}},$$

$$\mathbf{H}_{21} = -\Delta t \frac{\partial \mathbf{f}_D}{\partial \boldsymbol{\sigma}}, \qquad \mathbf{H}_{22} = 1 - \Delta t \frac{\partial \mathbf{f}_D}{\partial \mathbf{D}}.$$

[1]The symbols Δ and δ refer to incremental and iterative values, $\boldsymbol{\sigma}_n^{i+1} = \boldsymbol{\sigma}_n^i + \delta\boldsymbol{\sigma}_n^i$, $\Delta\boldsymbol{\sigma}_n^i = \boldsymbol{\sigma}_n^i - \boldsymbol{\sigma}_{n-1}$, where the sub- and superscripts refer to step and iteration numbers, respectively.

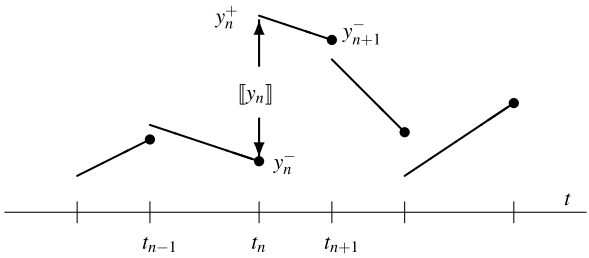


Fig. 24.1 The discontinuous Galerkin method, dG(1); notation

The algorithmic tangent matrix, i.e. the Jacobian of the algorithmic stress-strain relation has the simple form

$$\mathbf{C}^{\mathrm{ATS}} = \widetilde{\mathbf{H}}_{11}^{-1}\mathbf{C}, \tag{24.19}$$

where

$$\widetilde{\mathbf{H}}_{11} = \mathbf{H}_{11} - \mathbf{H}_{12}\mathbf{H}_{22}^{-1}\mathbf{H}_{21}.$$

As it can be seen, the Jacobian matrix is in general nonsymmetric due to the damage. The algorithmic tangent matrix is a necessity for the Newton method to obtain asymptotically quadratic convergence of the global equilibrium iterations.

24.4.2 The Discontinuous Galerkin Method

Rewrite the evolution equations (24.13) in the form

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}), \tag{24.20}$$

where $\mathbf{y} = [\boldsymbol{\sigma}^T, \mathbf{D}^T]^T$ and $\mathbf{f} = [\mathbf{f}_\sigma^T, \mathbf{f}_D^T]^T$. The discontinuous Galerkin method of degree q can be stated as follows [4]. For a given time interval $I_n = (t_n, t_{n+1}]$ find $\mathbf{y}$ (polynomial of degree q) such that

$$\int_{I_n} \big(\dot{\mathbf{y}} - \mathbf{f}(\mathbf{y})\big)^T \hat{\mathbf{y}} \mathrm{d}t + [\![\mathbf{y}_n]\!]^T \hat{\mathbf{y}}_n^+ = 0. \tag{24.21}$$

For the test functions $\hat{\mathbf{y}}$, polynomials of degree q are used. The notations $\mathbf{y}_n^+$ and $\mathbf{y}_n^-$ are the limits $\mathbf{y}_n^\pm = \lim_{\varepsilon\to 0} \mathbf{y}(t_n \pm |\varepsilon|)$, $[\![\mathbf{y}_n]\!] = \mathbf{y}_n^+ - \mathbf{y}_n^-$. These notations are illustrated in Fig. 24.1.

After the Newton linearisation step, the following system of linear equations is obtained:

$$\begin{bmatrix} A_{11}\mathbf{I} - \mathbf{M}_{11} & A_{12}\mathbf{I} - \mathbf{M}_{12} \\ A_{21}\mathbf{I} - \mathbf{M}_{21} & (1 + A_{22})\mathbf{I} - \mathbf{M}_{22} \end{bmatrix}^i \begin{Bmatrix} \delta\mathbf{y}^- \\ \delta\mathbf{y}^+ \end{Bmatrix}$$
$$= \begin{Bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{Bmatrix}^i - \begin{bmatrix} A_{11}\mathbf{I} & A_{12}\mathbf{I} \\ A_{21}\mathbf{I} & A_{22}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{y}^- \\ \mathbf{y}^+ \end{Bmatrix}^i - \begin{Bmatrix} \mathbf{0} \\ \mathbf{y}_n^{+i} - \mathbf{y}_n^- \end{Bmatrix}, \tag{24.22}$$

where

$$A_{ij} = \int_{I_n} N_i \dot{N}_j \mathrm{d}t, \qquad \mathbf{M}_{ij} = \int_{I_n} N_i \frac{\partial \mathbf{f}}{\partial \mathbf{y}} N_j \mathrm{d}t, \qquad \mathbf{r}_i = \int_{I_n} N_i \mathbf{f} \mathrm{d}t,$$

and N_i's are the linear interpolation functions $N_1 = (t - t_n)/\Delta t$, $N_2 = 1 - (t - t_n)/\Delta t$, which can be collected into a row vector $\mathbf{N} = [N_1, N_2]$. When the Newton iteration is converged after the k-th iteration, then $\mathbf{y}^-_{n+1} = (\mathbf{y}^-)^k$.

Partitioning the unknowns in the vector $\mathbf{y}$ as $\mathbf{y} = [(\boldsymbol{\sigma}^-)^T, (\boldsymbol{\sigma}^+)^T, \tilde{\mathbf{D}}^T]^T$, where $\tilde{\mathbf{D}} = [\mathbf{D}^{-T}, \mathbf{D}^{+T}]^T$, the coefficient matrix on the right-hand side of (24.22) can be written as

$$\mathbf{J}_{\mathrm{dG(1)}} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{G}_{1D} \\ \mathbf{B}_{21} & \mathbf{I} + \mathbf{B}_{22} & \mathbf{G}_{2D} \\ \mathbf{G}_{D1} & \mathbf{G}_{D2} & \mathbf{G}_{DD} \end{bmatrix},$$

where

$$\mathbf{B}_{ij} = A_{ij}\mathbf{I} - \mathbf{M}_{\sigma ij}, \qquad \mathbf{M}_{\sigma ij} = \int_{I_n} N_i \frac{\partial \mathbf{f}_\sigma}{\partial \boldsymbol{\sigma}} N_j \mathrm{d}t,$$

$$\mathbf{G}_{iD} = -\int_{I_n} N_i \frac{\partial \mathbf{f}_\sigma}{\partial \tilde{\mathbf{D}}} \mathbf{N} \mathrm{d}t, \qquad \mathbf{G}_{Di} = -\int_{I_n} \mathbf{N}^T \frac{\partial \mathbf{f}_D}{\partial \boldsymbol{\sigma}} N_i \mathrm{d}t,$$

$$\mathbf{G}_{DD} = \tilde{\mathbf{A}} - \int_{I_n} \mathbf{N}^T \frac{\partial \mathbf{f}_D}{\partial \tilde{\mathbf{D}}} \mathbf{N} \mathrm{d}t, \qquad \tilde{\mathbf{A}} = \begin{bmatrix} A_{11}\mathbf{I} & A_{12}\mathbf{I} \\ A_{21}\mathbf{I} & (1 + A_{22})\mathbf{I} \end{bmatrix}.$$

The Jacobian of the algorithmic stress-strain relation for the dG(1) method has the form

$$\mathbf{C}^{\mathrm{ATS}} = \left(\tilde{\mathbf{B}}_{11} - \tilde{\mathbf{B}}_{12}\tilde{\mathbf{B}}_{22}^{-1}\tilde{\mathbf{B}}_{21}\right)^{-1}\left(\mathbf{I} - \tilde{\mathbf{B}}_{12}\tilde{\mathbf{B}}_{22}^{-1}\right)\mathbf{C},$$

where

$$\tilde{\mathbf{B}}_{ij} = \mathbf{B}_{ij} - \mathbf{G}_{iD}\mathbf{G}_{DD}^{-1}\mathbf{G}_{Dj}.$$

From the results of the subsequent section, it seems that the dG(1) method performs well in computing inelastic material behaviour with damage. The only drawback is that the method is twice as laborious as the backward Euler scheme. However, numerical experiments show that the dG(1) scheme allows larger time steps to get a converged solution, such that the overall computing time can be even shorter than with the backward Euler method in strongly non-linear cases.

24.5 Numerical Example

24.5.1 Uniaxial Straining

Performance of the integrators is tested for the coupled viscous-damage model described in Sect. 24.3. For simplicity, the transition function is assumed to be unity

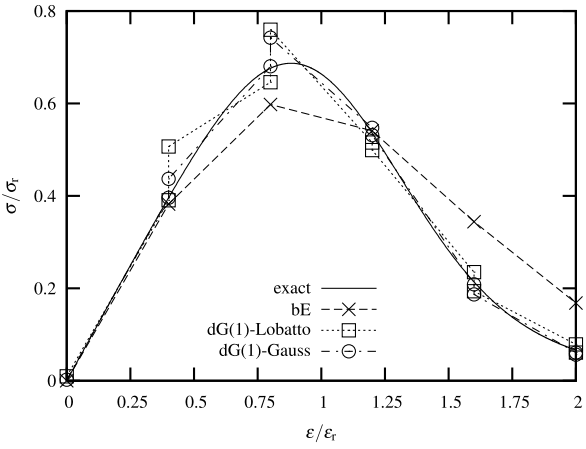


Fig. 24.2 The stress-strain relation for uniaxial constant strain-rate loading, from [3]

in this example, i.e. $\varphi_{\mathrm{tr}} \equiv 1$. The accuracy properties, when sufficiently large time steps are used, is of primary interest. The following material parameters are used: the Young modulus $E = 40\,\mathrm{GPa}$, reference stress $\sigma_{\mathrm{r}} = 20\,\mathrm{MPa}$, the viscosity parameters $\tau_{\mathrm{vp}} = 1000\,\mathrm{s}$, $\tau_{\mathrm{d}} = 0.2\,\mathrm{s}$, and the exponents $p = 4$ and $r = 1.5$. The reference value Y_{r} is chosen as in (24.12).

The stress-strain curves for an uniaxial constant strain rate $\dot{\varepsilon}_{\mathrm{c}} = 5 \times 10^{-4}\,\mathrm{s}^{-1}$ are shown in Fig. 24.2, where the true dG(1) solution, i.e. a discontinuous, piecewise linear approximation is depicted. To keep the figure readable, the end point solution values for the dG(1) methods are connected in Fig. 24.3, where the damage and inelastic strain are shown as a function of strain. Ten equal time steps are used for strain up to $4\varepsilon_{\mathrm{r}}$, thus $\Delta t = 0.4$ s. Inability of the backward Euler scheme to capture the damage evolution well is clearly visible in these figures. The "exact" solution shown in Figs. 24.2 and 24.3 is obtained by using the dG(1) method with the time step $\Delta t = 8 \times 10^{-4}$ s, resulting in 5000 steps in the range shown in Fig. 24.3. The estimated relative error for this solution is less than 10^{-5}.

24.6 Finite Element Simulations

24.6.1 Compression Test with the Scalar Damage Model

A compressed specimen ($(x, y, z) \in \Omega = (0, L) \times (0, B) \times (0, H)$, $L = 200\,\mathrm{mm}$, $B = 100\,\mathrm{mm}$, $H = 1\,\mathrm{mm}$) is analysed under a plane strain condition, as shown in Fig. 24.4. A'strain localisation into a shear band is expected to take place due to damage-induced strain softening. The horizontal displacement at the left-hand side edge is prescribed at a constant rate $\dot{u}_{\mathrm{prescibed}}$ and constrained to remain straight. A von-Mises type viscoplastic solid is used, i.e. $\bar{\sigma} = \sigma_{\mathrm{eff}}$. The constitutive parameters have the following values: the Young modulus E = 40 GPa, the Poisson ratio $\nu = 0.3$, reference stress σ_{r} = 20 MPa, the viscoplastic relaxation time

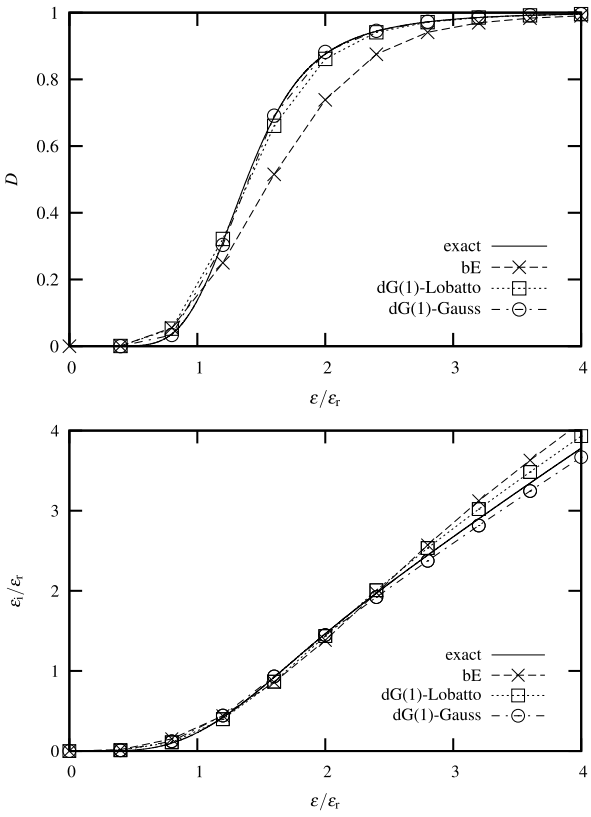


Fig. 24.3 Uniaxial constant strain-rate loading. For the dG(1) schemes, only the end points are connected [3]

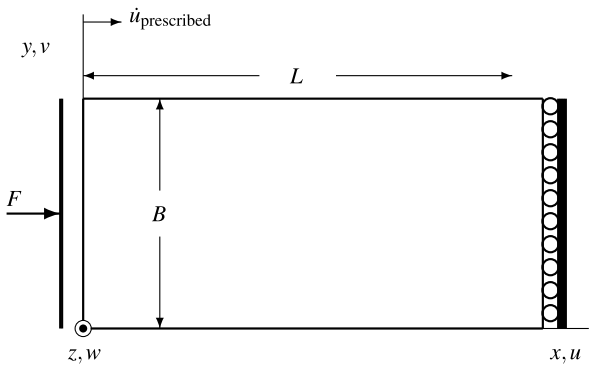
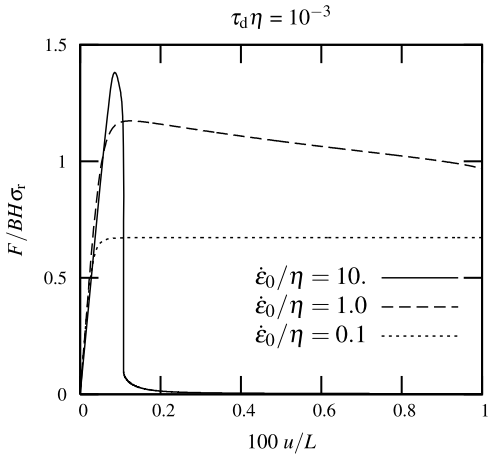


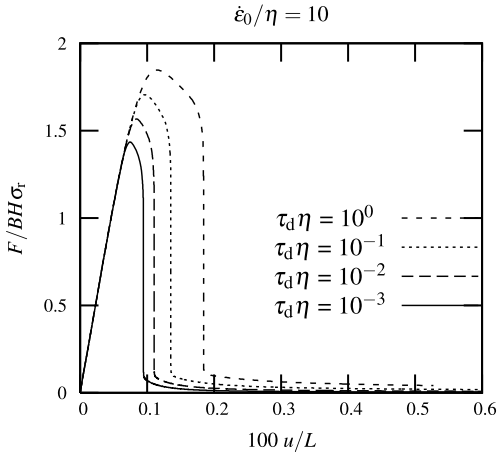
Fig. 24.4 The problem description

$\tau_{\text{vp}} = 1000\,\text{s}$, and the transition strain rate $\eta = 10^{-3}\,\text{s}^{-1}$. The exponents have the values: $p = n = 4$, and $r = 1.5$.[2]

[2]This corresponds to the same case as in [5], where the damage potential (24.9) was in the scalar case was defined in a slightly different way.



(a) The effect of the loading rate



(b) The effect of the "damage viscosity" τ_d in the brittle regime

Fig. 24.5 Load-displacement curves with the mesh of 12×6 elements

Eight-node-trilinear elements with the mean dilatation formulation [7] were used in the computations, which were carried out for two different meshes, a coarse mesh of 12×6 elements and a finer mesh of 48×24 elements. To trigger the unstable localisation, an imperfection via a small patch of elements was introduced by reducing the reference stress by 5 %.

Figure 24.5 shows the load-displacement curves calculated for three different loading rates (on the upper left) and four different damage relaxation times (on the upper right) using the coarse mesh, and for both meshes considering that $\tau_d\eta = 10^{-3}$ and $\dot{\varepsilon}_0/\eta = 10$ (at the bottom). The average strain rate is defined as $\dot{\varepsilon}_0 = \dot{u}_{\text{prescribed}}/L$. In comparison to the results of pure material behaviour (Fig. 24.3,

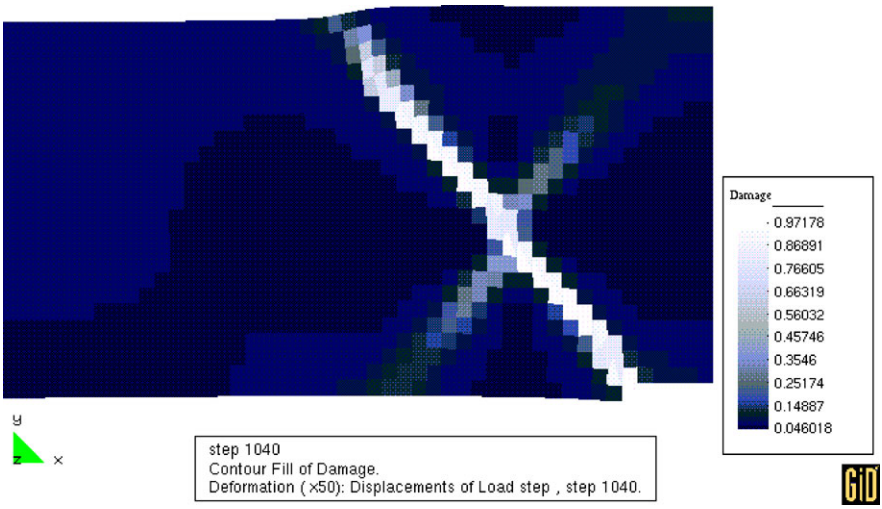


Fig. 24.6 Damage D distribution for $\dot{\varepsilon} = 10\eta$ and $\tau_d \eta = 10^{-3}$ at the end of the computation ($F = 0.618 B H \sigma_r$). A mesh of 48×24 elements. Displacements magnified 50 times

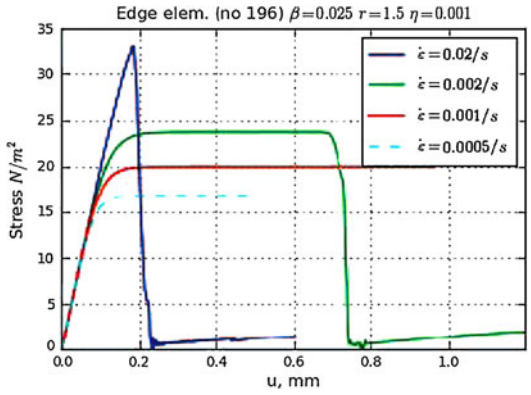
upper), the softening behaviour of the structure is much more rapid due to the localisation band.

As explained in the preceding section, a large number of time step reductions, due to diminished convergence of local iterations, had to be done during the computations, especially in the computations for the highest loading rate.

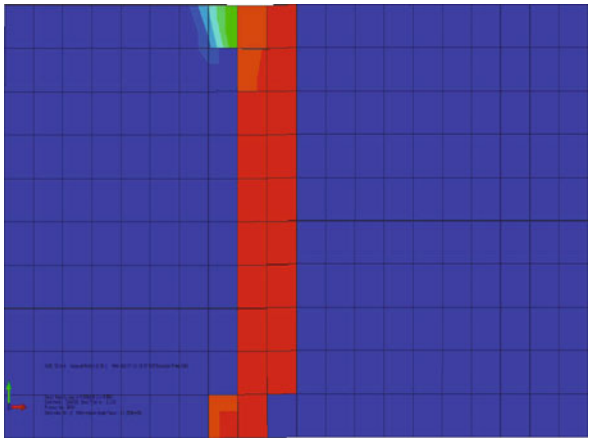
Damage distribution is shown in Fig. 24.6. It can be observed that damage bands are approximately at $\pm 45°$ angles as in the classical strain-softening von-Mises type elastoplasticity. Therefore it could be concluded that the scalar damage model is unable to capture the correct failure mode characteristic to brittle materials. It should be noted that the failure mode in tension is identical to the mode in compression with the scalar damage model. As it can be seen from the next section, to be able to predict the failure mode correctly, at least the vectorial damage model should be used.

24.6.2 Compression/Tensile Tests with the Vectorial Damage Model

The model with the vector description of damage has been implemented in the commercial finite element code ABAQUS as a user subroutine. A simple tensile test of the same specimen as in the previous example has been simulated using different loading rates, see Fig. 24.7. The same material parameters are used as with the scalar damage model simulation. The seed parameter for the damage initialisation has been $\beta = 0.025$. In Fig. 24.7(a) the load-displacement curves are shown with



(a)



(b)

Fig. 24.7 The tensile test: stress-displacement curves with different loading rates and localisation of damage in the brittle case $\dot{\varepsilon} > 0.001\ \mathrm{s}^{-1}$

different loading rates and the failure mode is shown in Fig. 24.7(b). It can be seen that the damage is localising in an area which has a width larger than one element layer.

For the compressive loading case, the damage vectors are shown in Fig. 24.8. As it can be seen, the splitting failure mode starts to develop from the weaker elements in the mesh.

24.7 Concluding Remarks

A phenomenological constitutive model for modelling the ductile-to-brittle transition due to an increased strain rate is presented. In the present model, the dissipation potential is additively split into damage and visco-plastic parts and the transition behaviour is obtained using a stress-dependent damage potential. In this study,

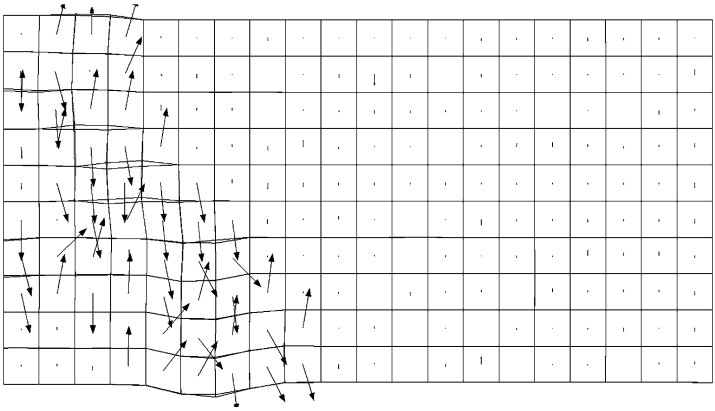


Fig. 24.8 The compression test: damage vectors **D** at the integration points on the softening regime

isotropic and vectorial damage coupled with von-Mises type viscoplastic flow are considered. However, the chosen approach allows easily an extension to more advanced damage models applicable also for realistic simulations of pressure dependent materials. To predict the correct failure mode for brittle solids, the damage cannot be described by a scalar variable. If the vectorial damage model is used, the tensile failure and splitting failure in compression can be simulated. Further investigations will be focused on the study of a material length scale.

The numerical implementation is also discussed. Due to the unstable nature of damage, the conventional backward Euler method does not perform well. Oscillations in the damage variable can result in convergence problems in the local Newton iteration at the integration point level. The discontinuous Galerkin approach seems to result in accurate results also for large time steps, and in addition, it seems to improve the convergence of the global equilibrium equations. Further studies will be directed to develop a robust integration scheme for inelastic constitutive models coupled with damage.

Acknowledgements This research has been supported in part by the Academy of Finland, decision number 121778.

References

1. Askes H, Hartikainen J, Kolari K, Kouhia R (2009) Dispersion analysis of a strain-rate dependent ductile-to-brittle transition model. In: Mäkinen R, Neittaanmäki P, Tuovinen T, Valpe K (eds) Proceedings of the 10th finnish mechanics days, University of Jyväskylä, Jyväskylä, pp 478–489
2. Duvault G, Lions L (1972) Inequalities in mechanics and physics. Springer, Berlin
3. Eirola T, Hartikainen J, Kouhia R, Manninen T (2006) Some observations on the integration of inelastic constitutive models with damage. In: Dahlblom O, Fuchs L, Persson K, Ristinmaa M, Sandberg G, Svensson I (eds) Proceedings of the 19th nordic seminar on computational mechanics, Division of Structural Mechanics, LTH, Lund University, pp 23–32

4. Eriksson K, Estep PHD, Johnsson C (1996) Computational differential equations. Studentlitteratur
5. Fortino S, Hartikainen J, Kolari K, Kouhia R, Manninen T (2006) A constitutive model for strain-rate dependent ductile-to brittle-transition. In: von Hertzen R, Halme T (eds) The IX finnish mechanics days, Lappeenranta University of Technology, Lappeenranta, pp 652–662
6. Frémond M (2002) Non-smooth thermomechanics. Springer, Berlin
7. Hughes T (1987) The finite element method. Linear static and dynamic finite element analysis. Prentice-Hall, Englewood Cliffs
8. Kolari K (2007) Damage mechanics model for brittle failure of transversely isotropic solids—finite element implementation. Tech rep 628, VTT Publications, Espoo
9. Kouhia R (2004) A time discontinuous Petrov-Galerkin method for the integration of inelastic constitutive equations. In: Neittaanmäki P, Rossi T, Majava K, Pironneau O (eds) ECCOMAS 2004 CD-ROM proceedings
10. Kouhia R, Marjamäki P, Kivilahti J (2005) On the implicit integration of rate-dependent inelastic constitutive models. Int J Numer Methods Eng 62(13):1832–1856
11. Lemaitre J (1992) A course on damage mechanics. Springer, Berlin
12. Lemaitre J, Chaboche J-L (1990) Mechanics of solid materials. Cambridge University Press, Cambridge
13. Perzyna P (1966) Fundamental problems in viscoplasticity. Advances in Applied Mechanics, vol 9. Academic Press, London
14. Ristinmaa M, Ottosen N (2000) Consequences of dynamic yield surface in viscoplasticity. Int J Solids Struct 37:4601–4622
15. Runesson K, Sture S, Willam K (1988) Integration in computational plasticity. Comput Struct 30:119–130
16. Simo J, Hughes T (1998) Computational inelasticity, 1st edn. Springer, New York
17. Wallin M, Ristinmaa M (2001) Accurate stress updating algorithm based on constant strain rate assumption. Comput Methods Appl Mech Eng 190:5583–5601