Modeling and Optimization in Science and Technologies

Mihnea Dulea · Aneta Karaivanova
Anastasis Oulas · Ioannis Liabotis
Danica Stojiljkovic · Ognjen Prnjat *Editors*

High-Performance Computing Infrastructure for South East Europe's Research Communities

Results of the HP-SEE User Forum 2012

HP-SEE
High-Performance Computing Infrastructure
for South East Europe's Research Communities

Springer

Modeling and Optimization in Science and Technologies

Volume 2

Series Editors

Srikanta Patnaik (Editor-in-Chief)
SOA University, Orissa, India

Ishwar K. Sethi
Oakland University, Rochester, USA

Xiaolong Li
Indiana State University, Terre Haute, USA

Editorial Board

Li Cheng,
Department of Mechanical Engineering,
The Hong Kong Polytechnic University,
Hong Kong

Jeng-Haur Horng,
Department of Power Mechnical Engineering,
National Formosa University,
Yulin,
Taiwan

Pedro U. Lima,
Institute for Systems and Robotics,
Lisbon,
Portugal

Mun-Kew Leong,
Institute of Systems Science,
National University of Singapore

Muhammad Nur,
Faculty of Sciences and Mathematics,
Diponegoro Unersity,
Semarang,
Indonesia

Kay Chen Tan,
Department of Electrical and Computer Engineering,
National University of Singapore,
Singapore

Yeon-Mo Yang,
Department of Electronic Engineering,
Kumoh National Institute of Technology,
Gumi, South Korea

Liangchi Zhang,
School of Mechanical and Manufacturing Engineering,
The University of New South Wales,
Australia

Baojiang Zhong,
School of Computer Science and Technology, Soochow University,
Suzhou, China

Ahmed Zobaa,
School of Engineering and Design,
Brunel University, Uxbridge,
Middlesex, UK

For further volumes:
http://www.springer.com/series/10577

Mihnea Dulea · Aneta Karaivanova
Anastasis Oulas · Ioannis Liabotis
Danica Stojiljkovic · Ognjen Prnjat
Editors

High-Performance Computing Infrastructure for South East Europe's Research Communities

Results of the HP-SEE User Forum 2012

Springer

Editors
Mihnea Dulea
Horia Hulubei National Institute
for Physics and Nuclear Engineering
Magurele
Romania

Ioannis Liabotis
GRNET S.A.
Greek Research and Technology Network
Athens
Greece

Aneta Karaivanova
Institute of Information and Communication
Technologies, Bulgarian Academy
of Sciences (IICT-BAS)
Sofia
Bulgaria

Danica Stojiljkovic
Institute of Physics Belgrade
University of Belgrade (IPB-UOB)
Belgrade
Serbia

Anastasis Oulas
Institute of Marine Biology, Biotechnology
and Aquacultures, Hellenic Centre for
Marine Research (IMBBC-HCMR)
Heraklion, Crete
Greece

Ognjen Prnjat
GRNET S.A.
Greek Research and Technology Network
Athens
Greece

ISSN 2196-7326 ISSN 2196-7334 (electronic)
ISBN 978-3-319-01519-4 ISBN 978-3-319-01520-0 (eBook)
DOI 10.1007/978-3-319-01520-0
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013944765

© Springer International Publishing Switzerland 2014
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

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

Preface

High-Performance Computing (HPC) is one of the key enablers of scientific research and of the development of Information Society in Europe. Enabling large-scale innovative research to be conducted through collaboration of distributed teams of scientists across the European Research Area paves the way towards a long-term vision of a sustainable, transparent, ubiquitous electronic infrastructure open to a wide range of scientific user communities. European Commission sees "strategic nature of High-Performance Computing as a crucial asset for the EU's innovation capacity, and calls on Member States, industry and the scientific communities, in cooperation with the Commission, to step up joint efforts to ensure European leadership in the supply and use of HPC systems and services by 2020." Pan-European PRACE infrastructure supports this strategy and currently provides the core of High-Performance Computing power for European research. In this context, the inclusion of less-developed regions of Europe, which suffer from the digital divide and brain-drain in all fields and especially high-technology, into the wider European Research Area is an aim closely aligned to this vision. HP-SEE project, co-funded by Framework Programme 7 of the European Commission, is currently advancing the computing infrastructures in South-East Europe (SEE), activating new user communities and enabling collaborative research across a number of fields, and thus contributes to closing the existing technological and scientific gap, and following the wider European HPC strategy.

The HP-SEE User Forum took place in Belgrade on October 17-19, 2012, at premises of National Library of Serbia and hosted by the Institutes of Physics Belgrade: this scientific conference gathered 65 participants from 16 countries in the region and beyond, with main focus on the fields of Computational Physics, Computational Chemistry, Life Sciences, Scientific Computing and HPC Systems and Network Operations. Conference programme comprised of seven invited lectures, both from related e-Infrastructure projects and HP-SEE scientific community, 26 contributed oral presentations and a poster session featuring 11 poster presentations. This edition of papers from the User Forum is comprised of 20 peer-reviewed papers: 7 from Computational Physics applications in HPC, 5 from Computational Chemistry, 3 from Life Sciences, and 5 from Scientific computing and HPC operations. The project as well as the wider SEE community has benefited greatly from the exchange of experiences of leading scientists in

the region in their use of High-Performance Computing technology to empower their research.

In my role of HP-SEE project manager, I would like to thank the authors, reviewers, editors, programme and organization committees and guest speakers for their contributions to this event.

April 2013

Dr. Ognjen Prnjat
HP-SEE Project Manager

Conference Topics

- Computational Physics
- Computational Chemistry
- Life Sciences
- Scientific Computing
- HPC Systems and Network Operation

Keynote Speakers

Regional eInfrastructure Development for South East Europes Research Communities
Ioannis Liabotis, GRNET, Greece

International HPC - Building bigger pyramide
Richard Kenway, NeSC, UK

LinkSCEEM2: Development of an HPC ecoSystem in the Eastern Mediterranean
Jens Wiegand, CaSToRC, Cyprus

The impact of GISELA Science Gateway (GSG) on the supported Latin America VRC's
Jesus Cruz Guzman, UNAM, Mexico

Numerical Study of Ultracold Quantum Gases: Formation of Faraday Patterns, Geometric Resonances, and Fragmentation
Antun Balaž, IPB, Serbia

Monte Carlo methods for Electron Transport: Scalability Study using HP-SEE infrastructure
Aneta Karaivanova, IICT-BAS, Bulgaria

Use of High Performance Computing in (Bio)Chemistry
Ivan Juranić, FCUB, Serbia

Organization

Organization Committee

Danica Stojiljković	IPB, Serbia
Aleksandar Belić	IPB, Serbia
Antun Balaž	IPB, Serbia
Dušan Vudragović	IPB, Serbia
Vladimir Slavnić	IPB, Serbia
Ioannis Liabotis	GRNET, Greece
Ognjen Prnjat	GRNET, Greece
Dimitra Kotsokali	GRNET, Greece
Nikola Grkić	IPB, Serbia
Milica Cvetković	IPB, Serbia

Programme Committee Chairs

Mihnea Dulea	IFIN-HH, Romania
Aneta Karaivanova	IICT-BAS, Bulgaria
Panayiota Poirazi	GRNET, Greece
Ognjen Prnjat	GRNET, Greece

Programme Committee Members

Aleksandar Belić	IPB, Serbia
Alexandru Nicolin	IFIN-HH, Romania
Anastas Mishev	UKIM, Macedonia
Antun Balaž	IPB, Serbia
Emanouil Atanassov	IICT-BAS, Bulgaria
Hrachya Astsatryan	IIAP NAS RA, Armenia
Ioannis Liabotis	GRNET, Greece
Klaus Klingmueller	CASTORC, Cyprus
Manthos G. Papadopoulos	IOPC, Greece
Miklos Kozlovszky	SZTAKI, Hungary

Neki Frasheri	PUoT, Albania
Nenad Vukmirović	IPB, Serbia
Péter Stefán	NIIF, Hungary
Petru Bogatencov	RENAM, Moldova
Ramaz Kvatadze	GRENA, Georgia

Referees

N. Anastopoulos	M. Ifti	A. Oulas
E. Atanassov	D. Jakimovski	Lj. Pejov
A. Balaž	A. Karaivanova	B. Rybakin
V. Baran	P. Korosoglou	M. Spassova
A. Belić	M. Kozlovszky	P. Stefán
A. Chatziioannou	I. Liabotis	S. Stefanov
N. I. Dodoff	A. Nicolin	N. G. Vassilev
R. Gábor	K. Nikas	N. Vukmirović

HP-SEE User Forum 2012 was organized by the Institute of Physics Belgrade under the umbrella of High-Performance Computing Infrastructure for South East Europes Research Communities (HP-SEE), a project supported by European Commission through EU FP7 under the “Research Infrastructures” action (contract number 261499).

The Editors and the HP-SEE project would like to acknowledge valuable help and support by the National Library of Serbia which hosted the conference.

Contents

Simulation of Electron Transport Using HPC Infrastructure in South-Eastern Europe

Emanouil Atanassov, Todor Gurov, and Aneta Karaivanova

Institute of Information and Communication Technologies, BAS
Acad. G. Bonchev St., Bl.25A, 1113 Sofia, Bulgaria
{emanouil,gurov,anet}@parallel.bas.bg

Abstract. In this work we present Monte Carlo simulation of ultrafast electron transport in semiconductors. We study the scalability of the presented algorithms using high-performance computing resources in South-Eastern Europe. Numerical results for parallel efficiency and computational cost are also presented. In addition we discuss the coordinated use of heterogeneous HPC resources for one and the same application in order to achieve a good performance.

Keywords: Electron transport, Monte Carlo algorithms, scalability, parallel efficiency, high-performance computations.

1 Introduction

The Monte Carlo Methods (MCMs) provide approximate solutions to a variety of mathematical problems by performing statistical sampling experiments on a computer [1–4]. They are based on the simulation of random variables whose mathematical expectations are equal to a given functional of the solution of the problem under consideration.

Many problems in a transport theory and related areas can be described mathematically by a second kind integral equation:

$$f = \mathbb{K}(f) + \phi, \tag{1}$$

where $\mathbb{K}$ is an integral operator. In general, the physical quantities of interest are determined by functionals of the type:

$$J_g(f) \equiv (g, f) = \int_G g(x) f(x) dx, \tag{2}$$

where the domain $G \subset \mathbb{R}^d$ and $\mathbb{R}^d$ is the d-dimensional Euclidean space. The functions $f(x)$ and $g(x)$ belong to any Banach space X and to the adjoint space X^*, respectively, and $f(x)$ is the solution of (1).

The mathematical concept of the MC approach is based on the iterative expansion of the solution of (1):

$$f_s = \mathbb{K}(f_{s-1}) + \phi, \;\; s = 1, 2, \ldots, \tag{3}$$

where s is the number of iterations. In fact (3) defines a Neumann series

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_1, © Springer International Publishing Switzerland 2014

$$f_s = \phi + I\!K(\phi) + \ldots + I\!K^{s-1}(\phi) + I\!K^s(f_0), \qquad s > 1\,,$$

where $I\!K^s$ means the s-th iteration of $I\!K$. If the corresponding infinite series converges then the sum is an element f from the space X which satisfies (1).

The replacement of f by the Neumann series in (2), gives rise to a sum of consecutive terms which are evaluated by a MC method with the help of random estimators.

We define a random variable ξ such that its mathematical expectation is equal to $J(f)$: $E\xi = J(f)$.

Then we can define a MC method

$$\overline{\xi} = \frac{1}{N}\sum_{i=1}^{N} \xi^{(i)} \xrightarrow{P} J_g(f), \tag{4}$$

where $\xi^{(1)}, \ldots, \xi^{(N)}$ are independent values of ξ and $\xrightarrow{P}$ means stochastic convergence as $N \longrightarrow \infty$. The rate of convergence is evaluated by the "law of the three sigmas", [1, 3]:

$$P\left(|\overline{\xi} - J_g(f)| < 3\frac{\sqrt{Var(\xi)}}{\sqrt{N}}\right) \approx 0.997.$$

Here $Var(\xi) = E\xi^2 - E^2\xi$ is the variance of the MC estimator. Thus, a peculiarity of any MC estimator is that the result is obtained with a statistical error [1, 3, 7]. As N increases, the statistical error decreases proportionally to $N^{-1/2}$.

Thus, there are two types of errors – systematic (a truncation error) and stochastic (a probability error) [7, 8]. The systematic error depends on the number of iterations of the used iterative method, while the stochastic error is related to the the probabilistic nature of the MC method. From (1) and (3) one can get the value of the truncation error. If $f_0 = \phi$ then

$$f_s - f = I\!K^s(\phi - f).$$

The relation (4) still does not determine the computational MC algorithm: we must specify the modeling function (called sampling rule) for the random variable ξ.

$$\Theta = F(\beta_1, \beta_2, \ldots,), \tag{5}$$

where $\beta_1, \beta_2, \ldots,$ are uniformly distributed random numbers in the interval (0, 1). It is known that pseudorandom number generators are used to produce such sequences of numbers. They are based on specific mathematical algorithms. Now both relations (4) and (5) define a MC algorithm for estimating $J_g(f)$. The case when $g = \delta(x - x_0)$ is of special interest, because it is used for calculating the value of f at x_0, where $x_0 \in G$ is a fixed point.

Every iterative algorithm uses a finite number of iterations s. In practice we define a MC estimator ξ_s for computing the functional $J_g(f_s)$ with a statistical error. On the other hand ξ_s is a biased estimator for the functional $J_g(f)$ with

stochastic and truncation errors [7, 8]. The number of iterations can be a random variable when an ε-criterion is used to truncate the Neumann series or the corresponding Markov chain in the MC algorithm.

In order to compute the physical quantities of interest, we simulate a large number of trajectories, following the backward Monte Carlo algorithmic scheme. The trajectory is abolished when the time step becomes sufficiently small. All the physical quantities of interest are approximated as averages of random variables that are sampled in the Monte Carlo algorithm, so that one numerical trajectory contributes to all the quantities of interest, for better efficiency. Since the trajectories are independent, there is large amount of available parallelism that is exploited via MPI and, if possible, CUDA. In order to achieve accurate estimates, when the evolution time becomes large enough, we need billions of sampled trajectories. This requirements together with the high number of parallel processes or threads, puts a strain on the quality of the random number generator library. When quasi-random numbers are used, then one should be careful how to distribute the computations so that the desired error rate is attained.

The presented numerical results are obtained using HP-SEE computing infrastructure. The regional HPC infrastructure incorporates state of the art HPC clusters and two supercomputers BlueGene/P. The HP-SEE resources are located in 8 HPC resource centers in the countries: Bulgaria, Hungary, Romania, FYROM and Serbia. The total peak performance of the HP-SEE computing resources is about 120 Tflops in double precision. The GPU resources currently achieve 12 Tflops in single precision. More information can be found at www.hp-see.eu. In order to cope with the heterogeneity in the HP-SEE infrastructure and the evolution of the mathematical algorithms the application is built using a modular structure, containing several different modules/libraries. While we use standard libraries for random number generation like SPRNG and CUDA CURAND, we use our in-house libraries for scrambled quasi-random number generators, [4–6]. The main computations are organized in a library too, where we have different computational kernels for the case of CPU vs GPU computations. We also separate the parallelization part, based on MPI.

The paper is organized as follows. In Section 2 the quantum-kinetic equation is derived from a physical model describing electron transport in quantum wires. An integral form of the equation is obtained by reducing the dimensionality of space and momentum coordinates. The MC approach and corresponding MC algorithm are presented in Section 3. The numerical results using Bulgarian HPC resources are discussed in Section 4. Summary and directions for future work are given in Section 5.

2 The Quantum Kinetic Integral Equation

In the general case a Wigner equation for nanometer and femtosecond transport regime is derived from a three equations set model based on the generalized Wigner function [10]. The complete Wigner equation poses serious numerical

challenges. Two limiting versions of the equation corresponding to simplified physical conditions are considered in few works, namely, the Wigner-Boltzmann equation [11] and the homogeneous Levinson (or Barker-Ferry) equation [12, 13]. These equations are analyzed with various MCMs using spherical and cylindrical transformations to reduce the dimensions in the momentum space [14, 15]. The computer power of the European Grid infrastructure (EGI) in some cases is used to investigate above problems [16–18].

Here we consider a highly non-equilibrium electron distribution which propagates in a quantum semiconductor wire [19]. The electrons, which can be initially injected or optically generated in the wire, begin to interact with three-dimensional phonons. This is third limiting case, where the electron-phonon interaction is described on the quantum-kinetic level by the Levinson equation [20, 21], but the evolution problem becomes inhomogeneous due to the spatial dependence of the initial condition. The direction of the wire is chosen to be z, the corresponding component of the wave vector is k_z. The electrons are in the ground state $\Psi(\mathbf{r}_\perp)$ in the plane normal to the wire, which is an assumption consistent at low temperatures. The initial carrier distribution is assumed Gaussian both in energy and space coordinates, and an electric field can be applied along the wire.

The integral representation of the quantum kinetic equation for the electron Wigner function f_w in this case has the form [22]:

$$f_w(z,k_z,t) = f_w(z - \frac{\hbar k_z}{m}t + \frac{\hbar F}{2m}t^2, k_z, 0) + \tag{6}$$

$$\int_0^t dt'' \int_{t''}^t dt' \int d\mathbf{q}'_\perp \int dk'_z \left[S(k'_z,k_z,t',t'',\mathbf{q}'_\perp)\times\right.$$

$$f_w\left(z - \frac{\hbar k_z}{m}(t-t'') + \frac{\hbar F}{2m}(t^2-t''^2) + \frac{\hbar q'_z}{2m}(t'-t''), k'_z, t''\right) -$$

$$\left.S(k_z,k'_z,t',t''\mathbf{q}'_\perp) f_w\left(z - \frac{\hbar k_z}{m}(t-t'') + \frac{\hbar F}{2m}(t^2-t''^2) - \frac{\hbar q'_z}{2m}(t'-t''), k_z, t''\right)\right],$$

where

$$S(k'_z,k_z,t',t'',\mathbf{q}'_\perp) =$$

$$\frac{2V}{(2\pi)^3}|G(\mathbf{q}'_\perp)\mathcal{F}(\mathbf{q}'_\perp, k_z - k'_z)|^2 \left[(n(\mathbf{q}')+1)\times\right.$$

$$\cos\left(\frac{\epsilon(k_z) - \epsilon(k'_z) + \hbar\omega_{\mathbf{q}'}}{\hbar}(t'-t'') + \frac{\hbar}{2m}Fq'_z(t'^2 - t''^2)\right) +$$

$$\left. n(\mathbf{q}')\cos\left(\frac{\epsilon(k_z) - \epsilon(k'_z) - \hbar\omega_{\mathbf{q}'}}{\hbar}(t'-t'') + \frac{\hbar}{2m}Fq'_z(t'^2 - t''^2)\right)\right].$$

Here, $f(z,k_z,t)$ is the Wigner function described in the $2D$ phase space of the carrier wave vector k_z and the position z, and t is the evolution time.

The electric force F depends on the electric field E as follows: $F = eE/\hbar$, where the electric field is along the direction of the wire, e being the electron charge and $\hbar$ - the Plank's constant.

$n_{\mathbf{q}'} = 1/(\exp(\hbar\omega_{\mathbf{q}'}/\mathcal{K}T) - 1)$ is the Bose function, where $\mathcal{K}$ is the Boltzmann constant and T is the temperature of the crystal, corresponds to an equilibrium distributed phonon bath.

$\hbar\omega_{\mathbf{q}'}$ is the phonon energy which generally depends on $\mathbf{q}' = \mathbf{q}'_\perp + q'_z = \mathbf{q}'_\perp + (k_z - k'_z)$, and $\varepsilon(k_z) = (\hbar^2 k_z^2)/2m$ is the electron energy.

$\mathcal{F}$ is obtained from the Fröhlich electron-phonon coupling by recalling the factor $i\hbar$ in the interaction Hamiltonian:

$$\mathcal{F}(\mathbf{q}'_\perp, k_z - k'_z) = -\left[\frac{2\pi e^2\omega_{\mathbf{q}'}}{\hbar V}\left(\frac{1}{\varepsilon_\infty} - \frac{1}{\varepsilon_s}\right)\frac{1}{(\mathbf{q}')^2}\right]^{\frac{1}{2}},$$

where (ε_∞) and (ε_s) are the optical and static dielectric constants. The shape of the wire affects the electron-phonon coupling through the factor

$$G(\mathbf{q}'_\perp) = \int d\mathbf{r}_\perp e^{i\mathbf{q}'_\perp \mathbf{r}_\perp} |\Psi(\mathbf{r}_\perp)|^2,$$

where Ψ is the ground state of the electron system in the plane normal to the wire. If the cross-section of the wire is chosen to be a square with side a than we obtain:

$$|G(\mathbf{q}'_\perp)|^2 = |G(q'_x)G(q'_y)|^2 = \left(\frac{4\pi^2}{q'_x a\left((q'_x a)^2 - 4\pi^2\right)}\right)^2 4\sin^2(aq'_x/2)\times$$

$$\left(\frac{4\pi^2}{q'_y a\left((q'_y a)^2 - 4\pi^2\right)}\right)^2 4\sin^2(aq'_y/2).$$

3 Monte Carlo Approach

The equation (6) can be rewritten in the form:

$$f_w(z, k_z, t) = f_w(z - z(k_z, t), k_z, 0) + \tag{7}$$

$$\int_0^t dt'' \int_{t''}^t dt' \int_G d^3\mathbf{k}' K_1(k_z, \mathbf{k}', t', t'') \times f_w\left(z + h(k_z, q'_z, t, t', t'', F), k'_z, t''\right) +$$

$$\int_0^t dt'' \int_{t''}^t dt' \int_G d^3\mathbf{k}' K_2(k_z, \mathbf{k}', t', t'') \times f_w\left(z + h(k_z, -q'_z, t, t', t'', F), k_z, t''\right),$$

where

$$z(k_z, t) = \frac{\hbar k_z}{m}t - \frac{\hbar F}{2m}t^2,$$

$$h(k_z, q'_z, t, t', t'', F) = -\frac{\hbar k_z}{m}(t - t'') + \frac{\hbar F}{2m}(t^2 - t''^2) + \frac{\hbar q'_z}{2m}(t' - t''),$$

$$K_1(k_z, \mathbf{k}', t', t'') == S(k'_z, k_z, t', t'', \mathbf{q}'_\perp) = -K_2(\mathbf{k}', k_z, t', t''),$$

and

$$\int_G d^3\mathbf{k}' = \int d\mathbf{q}'_\perp \int_{-Q_2}^{Q_2} dk_z.$$

The values of the physical quantities are expressed by the following general functional of the solution of (7):

$$J_g(f) = \int_0^{\mathcal{T}} \int_D g(z, k_z, t) f_w(z, k_z, t) dz dk_z dt. \tag{8}$$

Here we specify that the phase space point (z, k_z) belongs to a rectangular domain $D = (-Q_1, Q_1) \times (-Q_2, Q_2)$, and $t \in (0, \mathcal{T})$.

The function $g(z, k_z, t)$ depends on the quantity of interest. Here, we are going to estimate by MC approach the Wigner function (6), the wave vector (and respectively the energy) $f(k_z, t)$, and the density distribution $n(z, t)$. The last two functions are given by the integrals

$$f(k_z, t) = \int \frac{dz}{2\pi} f_w(z, k_z, t) \text{ and, } n(z, t) = \int \frac{dk_z}{2\pi} f_w(z, k_z, t).$$

The MC estimator for evaluating the functional (8) using backward time evolution of the numerical trajectories can be constructed in the following way:

$$\xi_s[J_g(f)] = \frac{g(z, k_z, t)}{p_{in}(z, k_z, t)} W_0 f_w(., k_z, 0) + \frac{g(z, k_z, t)}{p_{in}(z, k_z, t)} \sum_{j=1}^{s} W_j^{\alpha} f_w\left(., k_{z,j}^{\alpha}, t_j\right). \tag{9}$$

Here

$$f_w\left(., k_{z,j}^{\alpha}, t_j\right) = \begin{cases} f_w\left(z + h(k_{z,j-1}, k_{z,j-1} - k_{z,j}, t_{j-1}, t'_j, t_j, F), k_{z,j}, t_j\right), \\ f_w\left(z + h(k_{z,j-1}, k_{z,j} - k_{z,j-1}, t_{j-1}, t'_j, t_j, F), k_{z,j-1}, t_j\right) \end{cases}$$

where $\alpha = 1$, in the first case, and $\alpha = 2$ in the second one;

$$W_j^{\alpha} = W_{j-1}^{\alpha} \frac{K_{\alpha}(k_{z\,j-1}, \mathbf{k}_j, t'_j, t_j)}{p_{\alpha} p_{tr}(\mathbf{k}_{j-1}, \mathbf{k}_j, t'_j, t_j)}, \text{ where } W_0^{\alpha} = W_0 = 1, \ \alpha = 1, 2, \ j = 1, \ldots, s.$$

The probabilities $p_{\alpha}, (\alpha = 1, 2)$ are chosen to be proportional to the absolute value of the kernels in (6). The initial density $p_{in}(z, k_z, t)$ and the transition density $p_{tr}(\mathbf{k}, \mathbf{k}', t', t'')$ are chosen to be tolerant[1] to the function $g(z, k_z, t)$ and the kernels, respectively. The first point (z, k_{z0}, t_0) in the Markov chain is chosen using the initial density, where k_{z0} is the third coordinate of the wave vector $\mathbf{k}_0$. Next points $(k_{z\,j}, t'_j, t_j) \in (-Q_2, Q_2) \times (t_j, t_{j-1}) \times (0, t_{j-1})$ of the Markov chain:

$$(k_{z0}, t_0) \to (k_{z1}, t'_1, t_1) \to \ldots \to (k_{z\,j}, t'_j, t_j) \to \ldots \to,$$

where $j = 1, 2, \ldots, s$ do not depend on the position z of the electrons. They are sampled using the transition density $p_{tr}(\mathbf{k}, \mathbf{k}', t', t'')$ as we take only the z-coordinate of the wave vector $\mathbf{k}$. Note the time t'_j conditionally depends on the

[1] $r(x)$ is tolerant to $g(x)$ if $r(x) > 0$ when $g(x) \neq 0$ and $r(x) \geq 0$ when $g(x) = 0$.

selected time t_j. The Markov chain terminates in time $t_s < \varepsilon_1$, where ε_1 is a fixed small positive number called a truncation parameter.

In order to evaluate the functional (8) by N independent samples of the estimator (9), we define a Monte Carlo method

$$\frac{1}{N}\sum_{i=1}^{N}(\xi_s[J_g(f)])_i \xrightarrow{P} J_g(f_s) \approx J_g(f), \tag{10}$$

where $\xrightarrow{P}$ means stochastic convergence as $N \to \infty$; f_s is the iterative solution obtained by the Neumann series of (7), and s is the number of iterations.

The relation (10) still does not determine the computational algorithm. To define a MC algorithm we have to specify the initial and transition densities, as well the modeling function (or sampling rule). The modeling function describes the rule needed to calculate the states of the Markov chain by using uniformly distributed random numbers in the interval $(0, 1)$. In our case we use SPRNG library [9].

Here, the following transition density is chosen:

$$p_{tr}(\mathbf{k}, \mathbf{k}', t', t'') = p(\mathbf{k}'/\mathbf{k})p(t, t', t''),$$

where

$$p(t, t', t'') = p(t, t'')p(t'/t'') = \frac{1}{t}\frac{1}{(t - t'')}$$

and

$$p(\mathbf{k}'/\mathbf{k}) = c_1/(\mathbf{k}' - \mathbf{k})^2$$

(c_1 is the normalized constant). Thus, if we know t, the next times t'' and t' are computed by using the inverse-transformation rule.

The wave vectors $\mathbf{k}'$ are sampled in the following algorithm:

1. **Sample** a random unit vector $\omega = (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)$ as $\sin\theta = 2\sqrt{(\beta_1 - \beta_1^2)}$, $\cos\theta = 2\beta_1 - 1$, and $\varphi = 2\pi\beta_2$ where β_1 and β_2 are uniformly distributed numbers in (0, 1);
2. **Calculate** $l(\omega) = -\omega \cdot \mathbf{k} + (Q_2^2 + (\omega \cdot \mathbf{k})^2 - \mathbf{k}^2)^{\frac{1}{2}}$, where $\omega \cdot \mathbf{k}$ denotes a scalar product between two vectors;
3. **Sample** $\rho = l(\omega)\beta_3$, where β_3 is an uniformly distributed number in (0, 1);
4. **Calculate** $\mathbf{k}' = \mathbf{k} + \rho\omega$.

We note that we have to compute all three coordinates of the wave vector although we need only the third one. The choice of $p_{in}(z, k_z, t)$ depends on the function $g(z, k_z, t)$. The cases when

$$(i)\quad g(z, k_z, t) = \delta(z - z_0)\delta(k_z - k_{z,0})\delta(t - t_0),$$
$$(ii)\quad g(z, k_z, t) = \frac{1}{2\pi}\delta(k_z - k_{z,0})\delta(t - t_0),$$
$$(iii)\quad g(z, k_z, t) = \frac{1}{2\pi}\delta(z - z_0)\delta(t - t_0),$$

are of special interest, because they estimate the values of the Wigner function, wave vector and density distribution in fixed points.

4 Parallel Implementation and Numerical Results

The stochastic error for the (homogeneous) Levinson or Barker-Ferry models has order $O(\exp(c_2 t)\, N^{-1/2})$, where t is the evolution time and c_2 is a constant depending on the kernels of the obtained quantum kinetic equation [14, 15]. Using the same mathematical techniques as in [14], we can prove that the stochastic error of the MC estimator under consideration has order $O(\exp\left(c_3 t^2\right) N^{-1/2})$. The factor $\exp\left(c_3 t^2\right)$ contains the term t^2 because there is a double integration over the evolution time in the the quantum kinetic equation (6). The estimate shows that when t is fixed and $N \to \infty$ the error decreases, but for large t the factor $\exp\left(c_3 t^2\right)$ looks ominous. Therefore, the MC algorithm described above solves an NP-hard problem concerning the evolution time. The suggested importance sampling technique, which overcomes the singularity in the kernels, is not enough to solve the problem for long evolution time with small stochastic error. In order to decrease the stochastic error we have to increase N - the number of Markov chain realizations. For this aim, a lot of CPU power is needed for achieving acceptable accuracy at evolution times above 100 femtoseconds.

The suggested simulation was tested with new random number generators using permutations. Optimizations of transition density using genetic algorithm and acceptance-rejection methods were done. Initial scientific results for simulation of electron transport on quantum wires and are obtained.

It is known that the MC algorithms are perceived as computationally intensive and naturally parallel [23]. They can usually be implemented via the so-called dynamic *bag-of-work* model [24]. In this model, a large MC task is split into smaller independent subtasks, which are then executed separately. One process or thread is designated as "master" and is responsible for the communications with the "slave" processes or threads, which perform the actual computations. Then, the partial results are collected and used to assemble an accumulated result with smaller variance than that of a single copy. The inherent characteristics of MC algorithms and the dynamic *bag-of-work* model make them a natural fit for the parallel architectures.

Our numerical results are obtained using the following HPC platforms:

(i) The biggest HPC resource in Bulgaria is the supercomputer BlueGene/P which is deployed at the Executive Agency "Electronic Communications Networks and Information Systems". It has two racks with 2048 PowerPC 450 processors (32 bits, 850 MHz), 8192 processor cores and a total of 4 TB random access memory. The theoretical peak performance is 27.85 Tflops.

(ii) The other HPC platform is the HPC cluster deployed at the institute of information and communication technologies of the Bulgarian academy of sciences. This cluster consists of two racks which contain HP Cluster Platform Express 7000 enclosures with 36 blades BL 280c with dual Intel Xeon X5560 @ 2.8Ghz (total 576 cores), 24 GB RAM per blade. There are 8 storage and management controlling nodes 8 HP DL 380 G6 with dual Intel X5560 @ 2.8 Ghz and 32 GB RAM. All these servers are interconnected via non-blocking DDR Infiniband interconnect at 20Gbps line speed. The theoretical peak performance is 3.23 Tflops. The HPC cluster was upgraded with an HP SL390s G7 4U Lft Half

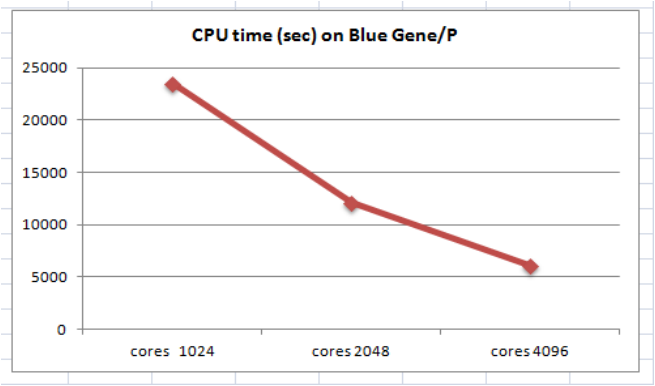


Fig. 1. Scalability on BlueGene/P when estimate Wigner function at $t = 180fs$ presented in the plane $z \times k_z$. The electric field is $15kV/cm$ and the number of Markov chains per pointsolution is 1 billion.

Tray Server with four NVIDIA Tesla M2090 6GB Modules, included in ProLiant SL6500 Scalable System Rack. The GPU cards have 2048 CUDA cores. The peak GPU computing performance exceeds the value of 2.66 Tflops in double precision or 5.32 Tflops in single precision. The GPU computing modules are connected to the HPCG blade cluster with QDR InfiniBand cards.

Both HPC resources are connected with 1 Gbps Ethernet fiber optics and all Bulgarian researchers can have access to them. Parallel programming paradigms supported by these HPC resources are Message passing, supporting several implementations of MPI: MVIAPICH1/2, OpenMPI, OpenMP.

By using the HPC resources we were able to reduce the computing time of the MC algorithm under consideration. The simulations of the Markov chain are parallelized on the the above HPC platforms by splitting the underlying random number sequences from the SPRNG library. In our research, the MC algorithm has been implemented in C++ language. The MPI implementation was MVIAPICH1.

The scalability results presented in Figures 1-2 are obtained for the problem with GaAs material parameters: the electron effective mass is 0.063, the optimal phonon energy is 36 meV, the static and optical dielectric constants are $\varepsilon_s = 12.9$ and $\varepsilon_\infty = 10.92$. The initial condition is a product of two Gaussian distributions of the energy and space. The k_z^2 distribution corresponds to a generating laser pulse with an excess energy of about 150 meV. The z distribution is centered around zero. The side of the wire is chosen to be 10 nanometers.

The values of the Wigner function $f(z, k_z, t)$ are estimated in a rectangular domain $(-Q_1.Q_1) \times (-Q_2, Q_2)$, where $Q_1 = 400$ nm and $Q_2 = 0.66nm^{-1}$ consisting of 800×260 points. The stochastic error for this case is relatively large. The relative mean squared error is in order of 10^{-3}.

The timing results for evolution time $t = 180fs$ and for all 800×260 points, are shown in Table 1. The number of the Markov chain's trajectories is 1 *billion*. The

Table 1. The CPU time (seconds) for all 800×260 points, the speed-up, and the parallel efficiency

Blades/Cores	CPU Time (s)	Speed-up	Parallel Efficiency
1 x 8 = 8	202300	-	-
4 x 8 = 32	50659	3.9937	0.99834
8 x 8 = 64	25423	7.9574	0.99467
16 x 8 = 128	12735	15.8853	0.99283
Blades/Cores/ Hyperthreading	CPU Time (s)	Speed-up	Parallel Efficiency
1 x 8 x 2 = 16	148602	-	-
4 x 8 x 2 = 64	37660	3.94588	0.98647
8 x 8 x 2 =128	18957	7.83889	0.97986
16 x 8 x 2 =256	9552	15.55716	0.97232

Table 2. The CPU time (seconds) for all 800×260 points, the speed-up, and the parallel efficiency

Cores	CPU Time (s)	Speed-up	Parallel Efficiency
1024	23498	-	-
2048	12082	1.9449	0.97245
4096	6091	3.8769	0.96923

results are obtained on the HPC cluster deployed at IICT-BAS. Tests with hyperthreading switched on and off were performed and the results show that the use of all logical cores, which are twice as many as physical cores, improves the speed with $33\% - 38\%$, which means that it is practical to turn hyperthreading on for these kinds of problems. In previous implementations of this code on desktop computers we have achieved higher improvements from hyperthreading. Thus we believe that the reason for the relatively small improvement is the improved efficiency of the current code, which means that more of the computations use floating point units of the processor and since these units are shared between threads, the hyperthreading can not offer more substantial gains. The parts of the code related to generation of pseudorandom numbers contain more integer operations and gain more from hyperthreading.

The results shown in Table 2 are obtained on IBM BlueGene/P. The solution again is estimated for evolution time $t = 180fs$ and for all 800×260 points, as the number of the Markov chain's trajectories again is 1 *billion*. Both timing results demonstrate a very good speed-up and parallel efficiency.

We also implemented our algorithm using CUDA and tested it on our GPU-based resources. The random number generator that we used was the default CURAND generator from the CUDA SDK. The parallelization of the code

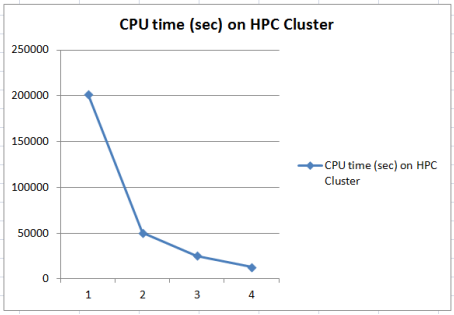


Fig. 2. Scalability on high-performance cluster

using CUDA was achieved without major rewrite of the code or changes to the program logic. The work is first split into blocks of trajectories to be computed. The master process sends the work to the slave processes, which initialize the respective GPU device and repeatedly execute the respective GPU kernel and return the results.

The same computation as above was performed in 67701 seconds on one NVIDIA M2090 card, which means that one card's performance is comparable to that of 3 blades with hyperthreading turned off. We believe that this result can be improved, because there could be some warp divergence due to logical statements in the code. This issue can be mitigated by changes in the way the samples are computed by the threads, to make sure that the divergence is limited. We also tested the algorithm when running on several GPU cards in parallel. When 6 Nvidia M2090 cards from the same server were used to compute 10^7 trajectories, we obtained about 93 % parallel efficiency. For such relatively small number of trajectories, the main source of inefficiency is the time spent in the cudaSetDevice call in the beginning of the computations.

5 Conclusions and Future Work

A quantum-kinetic model for the evolution of an initial electron distribution in a quantum wire has been introduced in terms of the electron Wigner function. The physical quantities, expressed as functionals of the Wigner function are evaluated within a MC approach. The developed MC method is characterized by the computational demands which are typical for quantum algorithms. The numerical results were obtained on two HPC platforms - IBM BlueGene/p and HP blade cluster. The test results show excellent parallel efficiency. The heterogeneous nature of the hardware resources that are available to the research team suggest the need to develop software that can use all those resources simultaneously. For such kinds of Monte Carlo or quasi-Monte Carlo codes it is possible to extend a dynamic load balancing scheme, that has already been developed by the team

for simpler problems. Since both the cluster and the supercomputer are positioned behind a firewall, a proxy mechanism will have to be employed in order to connect the super-master of the scheme with the working nodes and the user client machine. When the full information about the Wigner function is to be collected, there will be certain strain on the external network connectivity of the cluster or the supercomputer. Thus our next step will be the development and testing of the proxy module, that can also aggregate the results before sending them upstream. We have already developed code for parallel computations using several GPU cards, in a possibly heterogeneous configuration, but we believe we should concentrate to achieve better performance from the GPU version by some code refactoring to make the jumps more predictable.

Acknowledgment. This work was partially supported by the Bulgarian NSF grant DVCP02/1 CoE Supper CA++, by the European Commission under EU FP7 project HP-SEE (under contract number 261499).

References

1. Sobol, I.M.: Monte Carlo Numerical Methods, Nauka, Moscow (1973) (in Russian)
2. Kroese, D.P., Taimre, T., Botev, Z.I.: *H*andbook of Monte Carlo Methods. Wiley Series in Probability and Statistics. John Wiley and Sons, New York (2011)
3. Kalos, M.A., Whitlock, P.A.: Monte Carlo Methods, 2nd revised and enlarged edn. WILEY-VCH GmbH&Co. KGaA, Meinheim (2008)
4. Karaivanova, A.: Statistical Numerical Methods and Simulations, Demetra, Sofia (2012) (in Bulgarian)
5. Atanassov, E., Karaivanova, A., Ivanovska, S.: Tuning the Generation of Sobol Sequence with Owen Scrambling. In: Lirkov, I., Margenov, S., Waśniewski, J. (eds.) LSSC 2009. LNCS, vol. 5910, pp. 459–466. Springer, Heidelberg (2010)
6. Atanassov, E., Durchova, M.: Generating and Testing the Modified Halton Sequences. In: Dimov, I.T., Lirkov, I., Margenov, S., Zlatev, Z. (eds.) NMA 2002. LNCS, vol. 2542, pp. 91–98. Springer, Heidelberg (2003)
7. Mikhailov, G.A.: New Monte Carlo Methods with Estimating Derivatives, Utrecht, The Netherlands (1995)
8. Dimov, I., Dimov, T., Gurov, T.: A New Iterative Monte Carlo Approach for Inverse Matrix Problem. J. of Comp. and Appl. Math. 92, 15–35 (1998)
9. Scalable Parallel Random Number Generators Library for Parallel Monte Carlo Computations, SPRNG 1.0 and SPRNG 2.0, `http://sprng.cs.fsu.edu`
10. Nedjalkov, M., Kosik, R., Kosina, H., Selberherr, S.: A Wigner Equation for Nanometer and Femtosecond Transport Regime. In: Proceedings of the 2001 First IEEE Conference on Nanotechnology, pp. 277–281. IEEE, Maui (2001)
11. Nedjalkov, M., Kosina, H., Selberherr, S., Ringhofer, C., Ferry, D.K.: Unified Particle Approach to Wigner-Boltzmann Transport in Small Semiconductor Devices. Phys. Rev. B 70, 115319–115335 (2004)
12. Nedjalkov, M., Kosina, H., Selberherr, S., Ringhofer, C., Ferry, D.K.: Unified particle approach to Wigner-Boltzmann transport in small semiconductor devices. Physical Review B 70, 115319–115335 (2004)

13. Ringhofer, C., Nedjalkov, M., Kosina, H., Selberherr, S.: Semi-Classical Approximation of Electron-Phonon Scattering Beyond Fermi's Golden Rule. SIAM J. of Appl. Mathematics 64(6), 1933–1953 (2004) (the references therein)
14. Gurov, T.V., Whitlock, P.A.: An Efficient Backward Monte Carlo Estimator for Solving of a Quantum Kinetic Equation with Memory Kernel. Mathematics and Computers in Simulation 60, 85–105 (2002)
15. Gurov, T.V., Nedjalkov, M., Whitlock, P.A., Kosina, H., Selberherr, S.: Femtosecond Relaxation of Hot Electrons by Phonon Emission in Presence of Electric Field. Physica B 314, 301–304 (2002)
16. European Grid Infrastructure, `http://www.egi.eu/`
17. Atanassov, E., et al.: SALUTE application for Quantum Transport – New Grid Implementation Scheme. In: Proceedings of the Spanish Conference on e-Science Grid Computing, pp. 23–32 (2007)
18. Atanassov, E., Gurov, T., Karaivanova, A.: Ultra-fast Semiconductor Carrier Transport Simulation on the Grid. In: Lirkov, I., Margenov, S., Waśniewski, J. (eds.) LSSC 2007. LNCS, vol. 4818, pp. 461–469. Springer, Heidelberg (2008)
19. Nedjalkov, M., Gurov, T.V., Kosina, H., Vasileska, D., Palankovski, V.: Femtosecond Evolution of Spatially Inhomogeneous Carrier Excitations: Part I: Kinetic Approach. In: Lirkov, I., Margenov, S., Waśniewski, J. (eds.) LSSC 2005. LNCS, vol. 3743, pp. 149–156. Springer, Heidelberg (2006)
20. Levinson, I.: Translational invariance in uniform fields and the equation for the density matrix in the Wigner representation. Sov. Phys. JETP 30, 362–367 (1970)
21. Herbst, M., Glanemann, M., Axt, V., Kuhn, T.: Electron-phonon quantum kinetics for spatially inhomogeneous excitations. Physical Review B 67, 195305:1–195305:18 (2003)
22. Gurov, T., Atanassov, E., Dimov, I., Palankovski, V.: Femtosecond Evolution of Spatially Inhomogeneous Carrier Excitations: Part II: Stochastic Approach and GRID Implementation. In: Lirkov, I., Margenov, S., Waśniewski, J. (eds.) LSSC 2005. LNCS, vol. 3743, pp. 157–163. Springer, Heidelberg (2006)
23. Dimov, I., Tonev, O.: Monte Carlo Algorithms: Performance Analysis for Some Computer Architectures. J. of Comp. and Appl. Mathematics 48, 253–277 (1993)
24. Li, Y., Mascagni, M.: Grid-based Quasi-Monte Carlo Applications. Monte Carlo Methods and Appl. 11(1), 39–56 (2005)

Density Waves in Dipolar Bose-Einstein Condensates by Means of Symbolic Computations

Alexandru I. Nicolin[1,2,3] and Ionel Rata[1]

[1] Horia Hulubei National Institute for Physics and Nuclear Engineering,
30 Reactorului St., 077125 Magurele, Romania
[2] Faculty of Physics, University of Bucharest,
405 Atomistilor St., 077125 Magurele, Romania
[3] Faculty of Physics, West University of Timisoara,
4 Parvan St., Timisoara, Romania
alexandru.nicolin@nipne.ro

Abstract. We investigate by means of symbolic computations the emergence of density waves in cigar-shaped dipolar Bose-Einstein condensates and derive a series of approximate dispersion relations which exhibit the roton-maxon structure.

1 Introduction

Bose-Einstein condensates (BECs) with short-range interactions have become textbook material [1], with more than 100 groups world wide having achieved condensation of some atomic species and related theoretical efforts focused on a large set of research subjects which includes solitons, vortices, density waves and nonlinear phenomena at large [2–4]. Similar efforts have been focused on making efficient numerical and analytical recipes for the dynamics of ultra-cold quantum systems [6–10].

The landmarking experimental achievement of a chromium BEC in 2005 [11] and the recent condensation of dysprosium [12] bring forward fundamental question concerning the behavior of BECs with both short- and long-range interactions. The latter interactions stem from the dipolar interactions between atoms and can impact the dynamics of the condensate to the extent of stabilizing bright solitons (due to the anisotropy of the dipole-dipole interaction), changing the structure of vortex lattices and introducing a roton-maxon structure in the dispersion relation of density waves [13]. This last feature has attracted considerable attention on the theoretical side, with distinct investigations focused on the geometry of ground states configurations, abrupt transitions in Faraday patterns, enhanced sensitivity to external forcing, etc., but has so far eluded a direct experimental observation. In this paper we show by variational means using a cigar-shaped dipolar BEC with repulsive short-range interaction and effectively attractive long-range interactions that the roton-maxon structure is robust and that it can be recovered by analytical means using a standard computer algebra system. The backbone of our approach consists of a series of

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_2, © Springer International Publishing Switzerland 2014

symbolic calculations which allow us to compute directly the dipolar component of the interaction with various degrees of accuracy, thereby determining simple approximate formula for the roton-maxon spectrum. The rest of the paper is structured as follows: in Section 2 we present briefly the variational treatment of BECs, while in Section 3 we derive a series of novel approximate analytical results which exhibit the roton-maxon structure. Finally, Section 4 gathers our concluding remarks along with some suggestions for future research veins.

2 Variational Treatment

The variational treatment of BECs is textbook material (see Refs. [14, 15] for a detailed discussion) and amounts to computing the three-dimensional Lagrangian of condensate

$$L = \int_D \left\{ \frac{1}{2} \left(\psi \frac{\partial \psi^*}{\partial t} - \psi^* \frac{\partial \psi^*}{\partial t} \right) + \frac{1}{2} |\nabla \psi|^2 + \frac{g(t)N}{2} |\psi|^4 \right. \\ \left. + V(\mathbf{r}) |\psi|^2 + g_d |\psi|^2 \int_{\mathbb{R}^3} U(\mathbf{r} - \mathbf{r}') |\psi(\mathbf{r}')|^2 \right\}, \tag{1}$$

where $g(t) > 0$, $g_d > 0$ and $\psi = \psi(\mathbf{r})$ if not stated otherwise, and

$$U(\mathbf{r} - \mathbf{r}') = \frac{1 - \frac{3(z-z')^2}{(x-x')^2 + (y-y')^2 + (z-z')^2}}{\left((x-x')^2 + (y-y')^2 + (z-z')^2 \right)^{3/2}}, \tag{2}$$

with respect to a trial wave function. The ensuing result is minimized with respect to the variational parameters included in the trial wave function and a series of Euler-Lagrange ordinary differential equations is then obtained. For cigar-shaped condensates the prototypical wave function consists (for reasons of analytical tractability) of a Gaussian radial Ansatz on which a longitudinal periodic function is grafted such that we can investigate the emergence of density waves. The most convenient trial function uses a cos longitudinal function in which case one has to compute the Lagrangian in equation (1) using

$$\psi = \frac{1}{\pi \Omega} \sqrt{\frac{k}{2 + u(t)^2 + v(t)^2}} \exp\left(-\frac{x^2 + y^2}{2\Omega^2} \right) \\ \times (1 + (u(t) + iv(t)) \cos kz), \tag{3}$$

where D extends over the entire $x - y$ plane and over the segment $[-\pi/k, \pi/k]$ on the z axis. The trial wave function is normalized as

$$\int_0^\infty dr 2\pi r \int_{-\frac{\pi}{k}}^{\frac{\pi}{k}} dz \, |\psi(\mathbf{r})|^2 = 1. \tag{4}$$

Please note that, unlike the Ansatz usually used for condensates with short-range interactions, the above Ansatz considers a constant radial width, the radial

dynamics of the condensate being therefore discarded. This feature of our Ansatz is consistent with the one- and two-dimensional partial differential equations used to describe cigar- and pancake-shaped condensates [16]. Following some simple algebraic manipulations one can easily show that the first four terms of the Lagrangian are given by

$$L_1 = \frac{\hbar(u(t)\dot{v}(t) - v(t)\dot{u}(t))}{2 + u(t)^2 + v(t)^2}, \tag{5}$$

$$L_2 = \frac{\hbar^2 \left(2 + \left(u(t)^2 + v(t)^2\right)\left(1 + k^2\Omega^2\right)\right)}{2m\left(2 + u(t)^2 + v(t)^2\right)\Omega^2}, \tag{6}$$

$$L_3 = \frac{kN\left(8 + 3u(t)^4 + 8v(t)^2 + 3v(t)^4 + 6u(t)^2\left(4 + v(t)^2\right)\right)}{16\pi^2\left(2 + u(t)^2 + v(t)^2\right)^2\Omega^2} \tag{7}$$

and

$$L_4 = \frac{m\Omega^4}{2}. \tag{8}$$

Computing the last term is somewhat less trivial but one can show (see Ref. [17] for details) that the dipolar term is given by

$$L_5 = \int_{-\pi/k}^{\pi/k} dz \left(n_l(z) \int_{-\infty}^{\infty} dz' n_l(z') K(z - z')\right) \tag{9}$$

where

$$K(z - z') = \frac{\pi}{4\Omega}\left(2\sqrt{(z - z')^2}\Omega - \exp\left(\frac{(z - z')^2}{2\Omega^2}\right)\mathrm{Erfc}\left(\frac{\sqrt{(z - z')^2}}{\sqrt{2}\Omega}\right)\right) \tag{10}$$

and $n_l(z)$ is the longitudinal density profile of the condensate, namely

$$n_l(z) = \frac{1}{(\pi\Omega)^2}\frac{k(1 + (u(t) + iv(t))\cos kz)^2}{2 + u(t)^2 + v(t)^2}. \tag{11}$$

3 Roton-Maxon Structures

Computing explicitly the integral in L_5 is key in determining roton-maxon structures, but this cannot be done using $K(z - z')$ in equation (10). To this end, we will approximate the kernel $K(z - z')$ using simpler functions such that the aforementioned integral can be computed analytically on regular computer algebra systems. The first step in determining the approximate kernels is to note that

$$\lim_{z - z' \to 0} K(z - z') = -\frac{\pi^{3/2}\Omega}{2\sqrt{2}}, \tag{12}$$

$$\lim_{z-z'\to\infty} K(z-z') = 0 \tag{13}$$

and

$$\int_0^\infty da K(a) = -\frac{\pi^{3/2}\Omega^3}{\sqrt{2}}, \tag{14}$$

where $a = (z-z')^2$. These three relations capture the main features of the kernel, namely the value in the origin and at infinity and the norm, and we have used them in our search for approximate kernels which allow a straightforward analytic calculation of the dipolar term L_5. We have found that

$$K_1(z-z') = -\frac{\pi^{3/2}\Omega}{2\sqrt{2}} \exp\left(-(z-z')^{2\gamma} 2^{-\gamma} \left(\frac{\Gamma\left(1+\frac{1}{\gamma}\right)}{\Omega^2}\right)^{\gamma}\right) \tag{15}$$

where γ is real positive number and

$$K_2(z-z') = -\frac{\pi^{3/2}\Omega}{2\sqrt{2}} \left(1+(z-z')^2\right)^{-\frac{1+2\Omega^2}{2\Omega^2}}, \tag{16}$$

reproduce the main features of the original kernel and are simple enough such that L_5 can be computed without difficulty on standard computer algebra system such as MATHEMATICA, Maple, MATLAB (with Symbolic Math Toolbox) and Python (with the SymPy library).

For a condensate whose scattering length is modulated as $g(t) = g_0(1 + \epsilon\sin(\omega t))$ the dynamics of the density wave is given by a Mathieu equation of the form

$$\ddot{u}(t) + (a(k) + b(k)\sin(\omega t))u(t) = 0 \tag{17}$$

where $a(k)$ and $b(k)$ depend on k and the geometry of the system (e.g., L, the longitudinal extent of the condensate). The double integral in equation (9) can be computed most conveniently using MATHEMATICA's `Integrate` function and one easily finds using $K_1(z-z')$ that

$$a(k) = \frac{k^2\hbar}{2m}\left(-\exp\left(\frac{k^2\Omega^2}{2}\right)\frac{g_d k}{\pi\Omega^2\hbar} + \frac{k^2\hbar}{2m} + \frac{kNg(t)}{2\pi^2\Omega^2\hbar}\right) \tag{18}$$

for $\gamma = 1$,

$$a(k) = \frac{k^2\hbar}{2m\left(1+k^2\Omega^2\right)}\left(\frac{k^2\hbar}{2m} - \frac{\sqrt{2}g_d k}{\pi^{3/2}\Omega^2\hbar} + \frac{k^4\Omega^2\hbar}{2m} + \frac{k^3 Ng}{2\pi^2\hbar} + \frac{kNg}{2\pi^2\Omega^2\hbar}\right) \tag{19}$$

for $\gamma = 1/2$,

$$a(k) = \frac{k^2\hbar}{2m}\left\{\frac{k^2\hbar}{2m} + \frac{2^{3/2}g_d}{3^{2/3}k^{4/3}\Omega^{13/3}\hbar\pi^{1/2}}\exp\left(\frac{2}{9k^2\Omega^2}\right)\mathrm{Ai}\left(\frac{1}{3^{2/3}(k\Omega)^{4/3}}\right)\right.$$
$$\left. + \frac{2^{3/2}g_d(k\Omega)^{2/3}}{3^{1/3}k^{4/3}\pi^{1/2}\Omega^{13/3}\hbar}\exp\left(\frac{2}{9k^2\Omega^2}\right)\mathrm{AiPrime}\left(\frac{1}{3^{2/3}(k\Omega)^{4/3}}\right)\right\} \tag{20}$$

for $\gamma = 1/3$ and

$$a(k) = \frac{k^2\hbar}{2m}\left\{\frac{Nkg}{2\pi^2\Omega^2\hbar} - \frac{3^{1/4}g_d\cos\left(\frac{\sqrt{3}}{2k\Omega}\right)}{\sqrt{2k}\pi\Omega^{7/2}\hbar} + \frac{3^{1/4}\sqrt{2}g_d\cos\left(\frac{\sqrt{3}}{2k\Omega}\right)}{\sqrt{k}\pi\Omega^{7/2}\hbar}\mathrm{C}\left(\frac{3^{1/4}}{\sqrt{\pi k\Omega}}\right)\right.$$
$$\left. + \frac{k^2\hbar}{2m} - \frac{3^{1/4}g_d\sin\left(\frac{\sqrt{3}}{2k\Omega}\right)}{\sqrt{2k}\pi\Omega^{7/2}\hbar} + \frac{3^{1/4}\sqrt{2}g_d\sin\left(\frac{\sqrt{3}}{2k\Omega}\right)}{\sqrt{k}\pi\Omega^{7/2}\hbar}\mathrm{S}\left(\frac{3^{1/4}}{\sqrt{\pi k\Omega}}\right)\right\} \qquad (21)$$

for $\gamma = 1/4$. There is no general result for an arbitrary value of γ, but similar (though somewhat lengthier) results can be obtained for rational values of γ. For the second kernel we find that

$$a(k) = \frac{k^2\hbar}{2m}\left\{\frac{k^2\hbar}{2m} + \frac{kNg}{2\pi^2\Omega^2\hbar} - \frac{g_d 2^{1-\frac{1}{2\Omega^2}}k^{\frac{3}{2}+\frac{1}{2\Omega^2}}}{\pi\Omega\hbar\Gamma\left(\frac{1}{2\Omega^2}\right)}K_{\frac{1}{2}\left(1+\frac{1}{2\Omega^2}\right)}(k)\right\} \qquad (22)$$

which further reduces to

$$a(k) = \frac{k^2\hbar}{2m}\left\{\frac{k^2\hbar}{2m} + \frac{kNg}{2\pi^2\Omega^2\hbar} - \frac{g_d k\left(\sinh k - \cosh k\right)}{\sqrt{2\pi}\Omega^2\hbar}\right\} \qquad (23)$$

for large values of Ω.

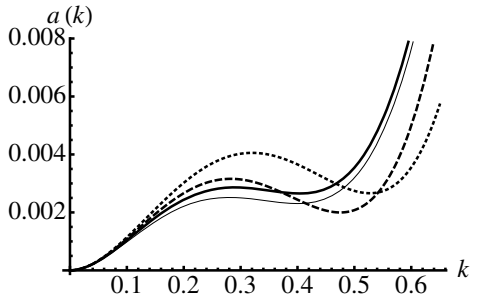


Fig. 1. A typical dispersion relation obtained with the approximate kernel $K_1(z-z')$ for $\hbar = m = \Omega = g_0 = 1$ and $\rho = Nk/2\pi = 1$. The thick dashed line corresponds to $\gamma = 1$ and $g_d = 3$, the thick dotted line corresponds to $\gamma = 1/2$ and $g_d = 4.2$, the thick full line corresponds to $\gamma = 1/3$ and $g_d = 6$ while the thin full line corresponds to $\gamma = 1/4$ and $g = 9$.

The main message conveyed by equations (18)-(23) is that the two approximate kernels reproduce analytically the roton-maxon structure seen in the dispersion spectrum of dipolar condensates, though for somewhat different parameters, thereby showing its robustness. To make this point transparent we plot in Fig. 1 typical dispersion relations for corresponding to the kernel $K_1(z-z')$ and in Fig. 2 similar dispersion relations for $K_2(z-z')$.

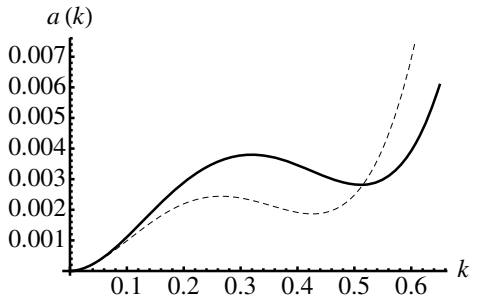


Fig. 2. A typical dispersion relation obtained with the approximate kernel $K_2(z - z')$ for $\hbar = m = \Omega = g_0 = 1$, $\rho = Nk/2\pi = 1$ and $g_d = 5.5$. The thick full line corresponds to equation (22) while the thin dashed line corresponds to equation (23).

4 Conclusions

We have determined by analytical means a series of approximate dispersion relations for density waves in cigar-shaped dipolar Bose-Einstein condensates with repulsive short-range interactions and effectively attractive long-range interactions which exhibit the roton-maxon structure. The novelty of these results comes from our self-consistent variational treatment and our accurate approximations of the dipolar kernel which allow us to simplify the dynamics of density waves in dipolar condensates to the level of ordinary differential equations. On the side of future research veins we mention the extension of these results for two-dimensional pancake-shaped condensates with attractive short-range interactions and effectively repulsive long-range ones and the emergence of density waves in binary dipolar condensates.

Acknowledgements. The authors thank Dumitru Mihalache and Mihnea Dulea for useful discussions. For this work A.I.N. was supported by CNCS-UEFISCDI through PN-II-ID-PCE-2011-3-0972, while I.R. was supported by ANCS through project PN 09370104/2013 and by the European Commission under EU FP7 project HP-SEE (under contract number 261499).

References

1. Pethick, C.J., Smith, H.: Bose-Einstein Condensation in Dilute Gases. Cambridge University Press, Cambridge (2008)
2. Kevrekidis, P.G., Frantzeskakis, D.J., Carretero-González, R. (eds.): Emergent Nonlinear Phenomena in Bose-Einstein Condensates. Springer, New York (2008)
3. Vidanovic, I., Balaz, A., Al-Jibbouri, H., Pelster, A.: Nonlinear Bose-Einstein-condensate dynamics induced by a harmonic modulation of the s-wave scattering length. Phys. Rev. A 84, 013618 (2011)
4. Balaz, A., Nicolin, A.I.: Faraday waves in binary nonmiscible Bose-Einstein condensates. Phys. Rev. A 85, 023613 (2012)

5. Vudragovic, D., Vidanovic, I., Balaz, A., Muruganandam, P., Adhikari, S.K.: C programs for solving the time-dependent Gross–Pitaevskii equation in a fully anisotropic trap. Comp. Phys. Comm. 183, 2021 (2012)
6. Vidanovic, I., Bogojevic, A., Belic, A.: Properties of quantum systems via diagonalization of transition amplitudes. I. Discretization effects. Phys. Rev. E 80, 066705 (2009)
7. Vidanovic, I., Bogojevic, A., Balaz, A., Belic, A.: Properties of quantum systems via diagonalization of transition amplitudes. II. Systematic improvements of short-time propagation. Phys. Rev. E 80, 066706 (2009)
8. Balaz, A., Vidanovic, I., Bogojevic, A., Belic, A., Pelster, A.: Ultra-fast converging path-integral approach for rotating ideal Bose-Einstein condensates. Phys. Lett. A 374, 1539 (2010)
9. Balaz, A., Vidanovic, I., Bogojevic, A., Belic, A., Pelster, A.: Fast converging path integrals for time-dependent potentials: I. Recursive calculation of short-time expansion of the propagator. J. Stat. Mech. P03004 (2011)
10. Balaz, A., Vidanovic, I., Bogojevic, A., Belic, A., Pelster, A.: Fast converging path integrals for time-dependent potentials: II. Generalization to many-body systems and real-time formalism. J. Stat. Mech. P03005 (2011)
11. Greismaier, A., Werner, J., Hensler, S., Stuhler, J., Pfau, T.: Bose-Einstein condensation of chromium. Phys. Rev. Lett. 94, 160401 (2005)
12. Lu, M., Burdick, M.Q., Youn, S.H., Lev, B.L.: Strongly dipolar Bose-Einstein condensate of dysprosium. Phys. Rev. Lett. 107, 190401 (2011)
13. Santos, L., Shlyapnikov, G.V., Lewenstein, M.: Roton-maxon spectrum and stability of trapped dipolar Bose-Einstein condensates. Phys. Rev. Lett. 90, 250403 (2003)
14. Pérez-García, V.M., Michinel, H., Cirac, J.I., Lewenstein, M., Zoller, P.: Low Energy Excitations of a Bose-Einstein Condensate: A Time-Dependent Variational Analysis. Phys. Rev. Lett. 77, 5320 (1996)
15. Yi, S., You, L.: Trapped condensates of atoms with dipole interactions. Phys. Rev. A 63, 053607 (2001)
16. Nath, R., Santos, L.: Faraday patterns in two-dimensional dipolar Bose-Einstein condensates. Phys. Rev. A 81, 033626 (2010)
17. Nicolin, A.I.: Density waves in dipolar Bose-Einstein condensates. Proc. Rom. Acad. A 14, 38 (2013)

Using Parallel Computing to Calculate Static Interquark Potential in LQCD

Dafina Xhako[1], Rudina Zeqirllari[2], and Artan Boriçi [2]

[1] Department of Engineering Science, Faculty of Professional Studies
"Aleksander Moisiu" University, Durres, Albania
dafinaxhako@yahoo.com
[2] Department of Physics, Faculty of Natural Sciences, University of Tirana
Tirana, Albania

Abstract. Lattice Quantum Chromodynamics (LQCD) is an algorithmic formulation of QCD, the mathematical model that describes quarks and their interactions. Computations in LQCD are typically very expensive and run on dedicated supercomputers and large computer clusters for many months. In this paper the calculations are performed in one of the clusters for supercomputing of HP-SEE (High-Performance Computing Infrastructure for South East Europe's Research Communities) project, that is located in Bulgaria (BG HPC). We use parallel computing with FermiQCD software, to determine the static quark-antiquark potential. In LQCD the static quark-antiquark potential can be derived from the Wilson loops. The standard method uses rectangular Wilson loops, while we test volume Wilson loops, using simulations with SU(3) gauge field configuration for different values of coupling constant and for different lattice sizes. The calculations are made for 100 statistically independent configurations, of gauge fields of the lattice.

Keywords: FermiQCD software, lattice, parallel computing, quark-antiquark potential, string tension, Wilson loops.

1 Introduction

Quantum Chromodynamics (QCD) is a theory of strong interaction, a fundamental force that describes the interactions between quarks and gluons. Analytic or perturbative solutions in low-energy QCD are difficult or impossible due to the highly nonlinear nature of the strong force. Wilson (1974) [1] introduced a non perturbative approximation based on discretization of space-time in a hypercubic finite lattice, with N- node per direction separated by a distance a.

In LQCD [2], [3], [4], fields representing quarks are defined at lattice sites while the gluon fields are defined on the links connecting neighboring sites. LQCD provides a framework for investigation of non-perturbative phenomena such as *confinement*. Numerical calculations in lattice using Monte Carlo methods are very expensive because of the space–time dimensions (4-dimensional problem). Progress in LQCD requires a combination of improvements in formulation, numerical techniques, and in

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_3, © Springer International Publishing Switzerland 2014

computer technology. The aim of this paper is the implementation and application of parallel computational techniques to study the properties of QCD in low energy regimes such as *quark confinement*. The solution is the use of parallel computational techniques to save time and costs in computations.

2 Materials and Methods

2.1 The Static Quark-Antiquark Potential

Since the original work by Creutz [5] in the SU(2) gauge theory, many computations of the potential from lattice gauge theories have been performed in SU(3), [6] [7], [8], [9]. Due to the asymptotic freedom of Yang-Mills theory, one expects at short distances a Coulomb-like behavior and the reliability of perturbation theory. At large distances quark confinement shows up and perturbative methods are no longer able to describe the behavior of physical observables; in this context, lattice gauge theories play an important role providing a full non perturbative approach.

The quark-antiquark potential can be extracted by large time behavior of the Wilson loops. For any closed path C with length n on the lattice, the Wilson loop is defined as the expectation value of the trace of the product of links around C,

$$W(C) = \langle tr \prod_{i=1}^{n} U_i \rangle . \tag{1}$$

The Wilson loops can be written as:

$$W(r,t) = \sum_{n \geq 1} C_n e^{-V_n(t)t} . \tag{2}$$

The effective potentials are calculated by

$$V(r)_{eff} = -\log \frac{W(r,t+1)}{W(r,t)} . \tag{3}$$

For a fixed r and for different values of time t, we select the value of effective potential while for long time t it is reached a plateau. Since $r = 1,..,6$ we have 6 values of effective potential. Finally, for different r we can fit the effective potential according to theoretical model in physical unit:

$$V(r) = V_0 + Kr + \frac{\alpha}{r}, \tag{4}$$

where V_0 is a constant, K is string tension parameter and α is a constant (the coefficient of the Coulomb-like term). Physically, this behavior implies the confinement of color in the flux tubes of the gluonic field. For large distances dominates the linear term Kr and for short distances dominates the Columbic term. In lattice units (non dimensional unit) equation (4) takes the form:

$$\hat{V}(r) = \hat{V}_0 + \hat{K}\frac{r}{a} + \frac{\alpha}{r}\hat{\alpha} , \tag{5}$$

where $\hat{V} = aV$, $\hat{V}_0 = aV_0$, $\hat{K} = a^2 K$ and $\hat{\alpha} = \alpha$.

For determining the scale of theory we have used the new method from Sommer's relation, with r_0=*0.5 fm*:

$$r_0{}^2 F(r_0) = 1.65\,, \tag{6}$$

where the lattice parameter is:

$$a = r_0 \sqrt{\frac{\tilde{K}}{\tilde{\alpha}+1.65}}\,. \tag{7}$$

To take physical quantity in continuum we repeat the simulation for different lattice volumes (taking physical length constant ~ *L=1.6fm*) and extrapolate in continuous limit when a→0.

2.2 FermiQCD and BG – HPC Cluster

FermiQCD [10], [11] is a library for fast development of parallel applications for Lattice Quantum Field Theories and Lattice Quantum Chromodynamics. It was designed to be easy to use, with a syntax very similar to common mathematical notation, and, at the same time, optimized for PC clusters. All FermiQCD algorithms are parallel, but parallelization is hidden from the high level programmer. FermiQCD components range from low level linear algebra, fitting and statistical functions to high level parallel algorithms specifically designed for lattice quantum field theories such as the Wilson [12], the Asqtad [13] action for KS fermions, the Domain Wall action [14] etc. One of the main differences between FermiQCD and libraries developed by other collaborations is that it follows an object oriented design as opposed to a procedural design.

This work makes use of results of the High-Performance Computing Infrastructure for South East Europe's Research Communities (HP-SEE), a project co-funded by the European Commission (under contract number 261499) through the Seventh Framework Programme. HP-SEE deals with multi-disciplinary international scientific communities (computational physics, computational chemistry, life sciences, etc.) stimulating the use of regional HPC infrastructure and its services.

The cluster for supercomputing that is located in Bulgaria (BG HPC) has SUSE Linux Enterprise Server 10 and compilers: IBM XL C/C++ Advanced Edition for Blue Gene/P V9.0 and for Blue Gene/P V11.1; GNU Tool chain (gcc, glibc, binutlils, gdb, python). It has 36Intel Xeon X5560 @2.8Ghz; 24 GB per node 576 cores; DDR Infiniband 2.5μs 20 Gbps; Small cluster with 4 Nvidia GTX 295 cards, total 1920 GPU cores also connected. Full information is available at [15].

Scalability Test of FermiQCD. We first tested the FermiQCD software deployment and allocation of computer time and simulation. We test it for different lattice sizes. The computation time falls exponentially (for example for lattice volumes 16^4).

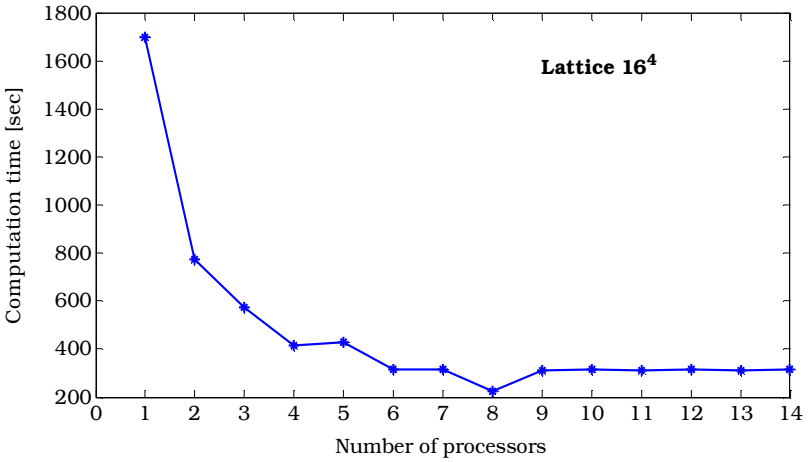


Fig. 1. The computation time from number of processors for lattice 16^4

Speedup and Efficiency Test. Let T(n,1) be the run-time of the fastest known sequential algorithm and let T(n,p) be the run-time of the parallel algorithm executed on p processors, where n is the size of the input (lattice volume). The speedup is then defined as $S(p) = \frac{T(n,1)}{T(n,p)}$,

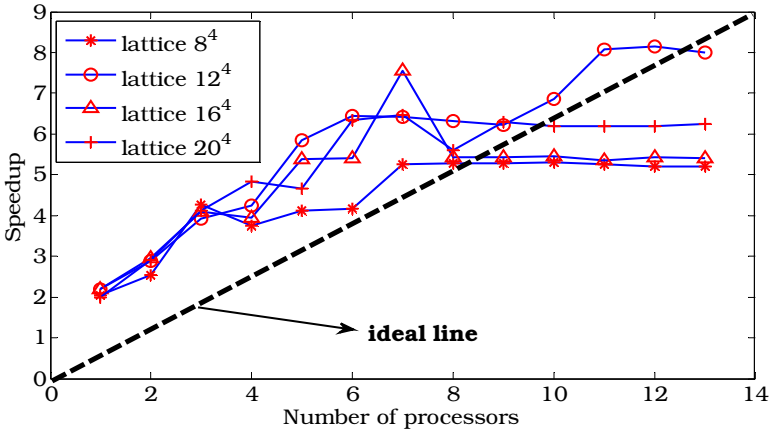


Fig. 2. Speedup form number of processors for different lattice volumes

The ideal speed up will be S(p) = p, so if we double for example the number of processors, will double the time of execution. Another metric to measure the performance of a parallel algorithm is efficiency, E(p), defined as: $E(p) = \frac{T(n,1)}{p\,T(n,p)} = \frac{S(p)}{p}$.

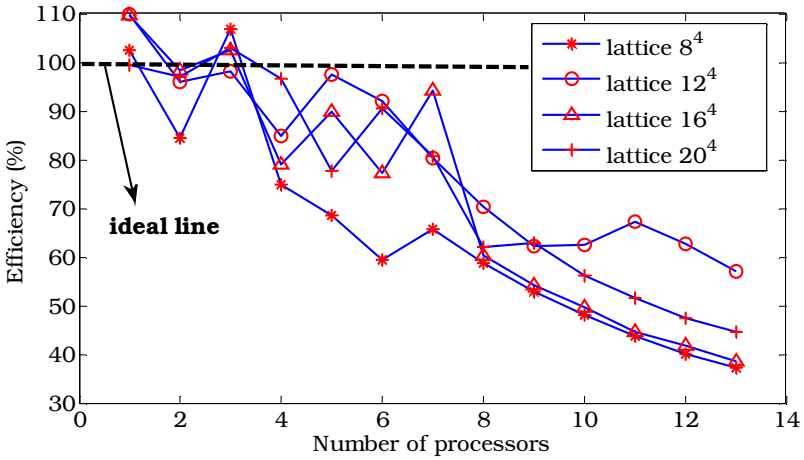


Fig. 3. Efficiency (in percentage) form the number of processors for different lattice volume

3 Preliminary Physics Results

3.1 Quark-Antiquark Potential from *Planar* and *Volume* Wilson Loops

We started from the example that estimates a single Wilson loop in FermiQCD and adapted it to calculate 36 rectangular *planar* Wilson loops for different values of time (t) and direction (r) on a lattice ($r = t = 1,..6$), and after that for *volume* Wilson loops *r1 x r2 x t* with *r1=r2=t=1,..6*. We have made simulations for 100 statistically independent configurations, for lattice 8^4, 12^4, 16^4, (lattice volume N^4), taking the physical volume (L4=(aN)^4) as constant. For each simulation we have changed the coupling constant $\beta = 6/g^2$ in order to keep constant physical volume, from [16].

Table 1. Lattice distance *a*, string tension with their statistical errors for quenched simulation with 8^4, 12^4, 16^4 (planar loops)

N^4	B	*a* from parameterization	*a* Calculated	String tension a^2K	Statistical error of *a*	Statistical error of a^2K
8^4	5.7	0.1707	0.2221(83)	0.3009 (39)	9.6796e-05	3.4427e-02
12^4	5.85	0.1230	0.1837(57)	0.16702(81)	3.4302e-05	1.0555e-02
16^4	6	0.0931	0.1060(69)	0.0596(12)	6.1446e-05	8.0894e-03

Preliminary result: Statistical error of *a*, is very small to justify the difference between *a_ calculated* and *a* from parameterization (it would be of the range ~ 10^{-1}).

Statistical processing of data's have not been performed on BG HPC cluster, but in our personal computers, in Matlab. We have written the scripts that calculate effective potentials, the coefficients V_0, K, alpha, statistical errors with Jackknife method, lattice parameter; graph of quark-antiquark from distance between them.

Table 2. Lattice distance *a*, string tension with their statistical errors for quenched simulation with 8^4, 12^4, 16^4 (volume loops)

N^4	β	*a* from parameterization	*a* calculated	String tension a^2K	Statistical error of *a*	Statistical error of a^2K
8^4	5.7	0.1707	0.1295(23)	0.08469(54)	0.9138e-03	0.0503
12^4	5.85	0.1230	0.0990(19)	0.03620(39)	0.4985e-03	0.0339
16^4	6	0.0931	0.0575(99)	0.0217(99)	0.0814e-03	0.0066

Preliminary result: Statistical error of a, is also like in case of planar loops, small to justify the difference between *a_ calculated* and *a* from parameterization (it would be of the range ~ 10^{-2}).

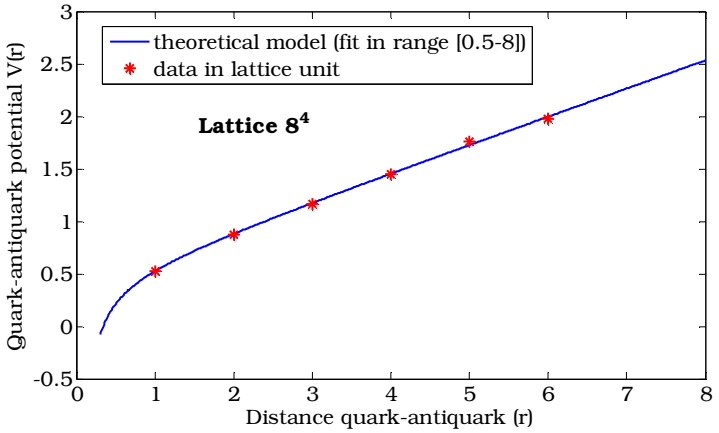


Fig. 4. Quark-antiquark potential (lattice 8^4) in lattice unit from *planar* Wilson loops

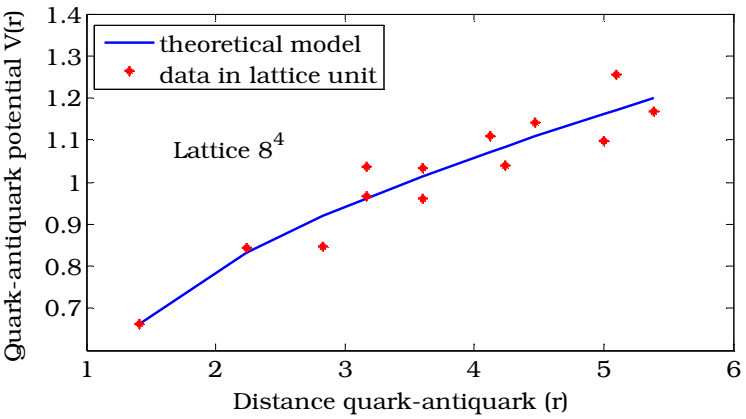


Fig. 5. Quark-antiquark potential (lattice 8^4) in lattice unit from *volume* Wilson loops

4 Conclusions

Techniques of parallel computing are effective in lattice QCD calculations. FermiQCD is one of the best programs that we can use in QCD to implement parallel computing. It has e very good scalability. Our first objective was to test FermiQCD program and techniques of parallel computing calculating the static quark-antiquark potential. Our graph results shows that the quark- anti-quark potential confirms a very important properties of QCD in low energy regimes such as *quark confinement.*

The statistical manipulation made in Matlab shows that the method using effective potential from planar Wilson loops (Fig.4) is better than volume Wilson loops (Fig.5), but we have to check the calculation of statistical errors of lattice spacing in our future work. As we can see form Tables 1 and 2 this error is very small to justify the difference between *a_ calculated* and *a* from parameterization. In our future work we have to increase the number of configurations, so increasing the statistical data we can take better results. New codes that we have written in FermiQCD are our contribution in this project of LQCD.

References

1. Wilson, K.G.: Phys. Rev. D 10, 1445 (1974)
2. Gupta, R.: Introduction to Lattice QCD. Lectures given at the LXVIII Les Houches Summer "School Probing the Standard Model of Particle Interactions", July 28-September 5, 150 pages, arXiv:hep-lat/9807028 (1997)
3. Gattringer, C., Lang, B.C.: Quantum Chromodynamics on the Lattice, 343 pages. Springer (2009) ISBN: 3642018491
4. Creutz, M.: Quarks, gluons and lattices. Cambridge Univ. Press, Cambridge (1983)
5. Creutz, M.: Phys. Rev. D 47, 661 (1993)
6. Schilling, K., Bali, G.S.: Int. J. Mod. Phys. C 4, 1167–1177 (1993)

7. Deldar, S.: Phys. Rev. D 62, 34509 (2000)
8. Bali, G.: Phys. Rev. D 62, 114503 (2000)
9. Parisi, G., Petronzio, R., Rapuano, F.: Phys. Lett. B 128, 418 (1983)
10. Di Pierro, M.: Matrix Distributed Processing and FermiQCD. In: Proceedings of the Workshop on Advanced Computing and Analysis Techniques, Fermilab, Batavia, IL 60510, October 16-20. Fermi National Accelerator Laboratory, USA (2000) arXiv:hep-lat/0011083
11. Di Pierro, M.: FermiQCD: A tool kit for parallel lattice QCD applications. Nucl. Phys. B. Proc. Suppl. 106, 1034–1036 (2002) arXiv:hep-lat/0110116
12. Wilson, K.G.: Confinement of Quarks. Phys. Rev. D 10, 2445–2459 (1974)
13. Bernard, C., et al.: MILC Collaboration, Quenched hadron spectroscopy with improved staggered quark action. Phys. Rev. D 58, 014503 (1998)
14. Kaplan, D.B.: A Method for simulating chiral fermions on the lattice. Phys. Lett. B 288(3), 342–347 (1992)
15. http://www.hp-see.eu
16. Guagnelli, M., Sommer, R., Witti, H.: Precision computation of a low energy reference scale in quenched lattice QCD. Nucl. Phys. B 535, 389–402 (1998)

Modelling of Disaster Spreading Dynamics

Igor Stanković, Milan Žeželj, Jelena Smiljanić, and Aleksandar Belić

Scientific Computing Laboratory, Institute of Physics Belgrade,
University of Belgrade, Pregrevica 118, 11080 Belgrade, Serbia
igor.stankovic@ipb.ac.rs

Abstract. Natural hazards are significant problem that every year cause important loses. We report on both theoretical models and simulations aimed at better understanding of disaster spreading in various networks. The structure of the networks in this work is obtained either through neighbor analysis in real space or using models which reproduce generic features of real networks (i.e., power, telecommunication, or road networks). Our investigations are focused on the understanding of interaction between network structure and disaster spreading mechanism. The probability that fire will propagate through fire protection strip is investigated and a model is introduced based on finite-size considerations in percolation theory. Also, the uncertainty in prediction of fire propagation rate due to the local inhomogeneities of the vegetation cover is investigated. Finally, uncertainty in cascade failures of network infrastructure is analyzed for a model where edges have limited capacity. The importance of the results for disaster prevention and control is discussed as well.

1 Introduction

The design of prevention measures and distribution of resources needed for efficient response to disaster event is a challenging problem. Examples include control of fire propagation [1], power transmission grid failure [2], information loss in communication systems [3, 4], and traffic congestions [5]. Development of new strategies for infrastructure failure prevention and damage control, such as emergency response and recovery call for external resources, which are limited, is important topic with many practical implications in real systems. A considerable research effort is underway to improve understanding of interaction of the structure and disaster propagation which considerably contributes to the understanding of processes going on in these networks. Experimental studies of disasters under real conditions are performed but understandably at scales much smaller then real disasters [7]. it is not realistic to expect allocation of resources for systematic data gathering during the disasters. Therefore mathematical and computer models are often very helpful tools to extend human knowledge beyond limited input data. However, the complexity of systems struck by disasters does not allow one to model the interactions of all involved entities and processes in detail and especially not in real time. Therefore, we have to capture them by an appropriate generic model. During 80's a line of research was initiated in which

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_4, © Springer International Publishing Switzerland 2014

model were used that reproduce and explain vegetation recovery and fire spread distribution in quite abstract way [8, 10]. These lattice models were used to describe fire size distributions which have a power law form [9, 10].

In this proceeding we report on our recent efforts to develop both theoretical models and simulations aimed on better understanding of disaster spreading and control in various networks. The structure of the networks is obtained either through neighbor analysis in real space or using models which reproduce generic features of real networks (i.e., power, telecommunication, or road networks) [3, 11–13]. The percolation and transport models on networks can be used to evaluate the disaster probability and impact. This information is important in practice for deciding how to effectively distribute resources in order to fight disasters the best. For example, finite size scaling considerations in percolation theory can be directly applied to design fire spreading prevention routes. In case of the fire behavior predictions uncertainty comes from variability in weather and fuel (vegetal) state inputs. Commonly model assumptions, such as fuel homogeneity and steady-state spread, limit the capability of the fire propagation models to provide reliable results, possibly leading to uncertainties of unknown magnitude [14]. We introduce a simple off-lattice deterministic reaction-thermal diffusion model in order to describe the dynamical evolution of the fire. Introduced model is not so specific as those employed by the ecologists but still introduces fire activation energy and heat dissipation. The model has an advantage that it can be applied in real time. We demonstrate how this simple model can provide interesting information about dynamics of the forest fire propagation. In the final part of proceeding we present a model of cascading failure. Cascading failure can happen in many infrastructure networks, including the electrical power grid, the Internet, road systems, and so on. The phenomenon is refereed to us an avalanching type of process, where the failure of a single or of a few network components can result in a large-scale breakdown of the network. In recent years cascading breakdown in complex networks has received considerable attention [2, 15–21]. Most previous existing works on cascading failures only focused on attacks on nodes rather than on edges. Attacks on edges are as important for the network security as those on nodes, and therefore deserve a careful investigation. In this proceeding, we analyze the cascade of overload failures in complex networks, where a edge failure and a subsequent network-wide redistribution of loads might trigger further cascading failures.

The proceeding is organized as follows: in Sec.2 we investigate the fire percolation probability dependence on density and shape of fire control paths. Based on finite size considerations in percolation theory we derive a simple model for assessment of fire propagation risk through corridor. In Sec.3, deterministic reaction-thermal diffusion model is introduced and simulation results for different vegetal phase densities are presented. Cascade of overload failures in complex networks are analyzed in Sec.4. Conclusion is given in Sec.5.

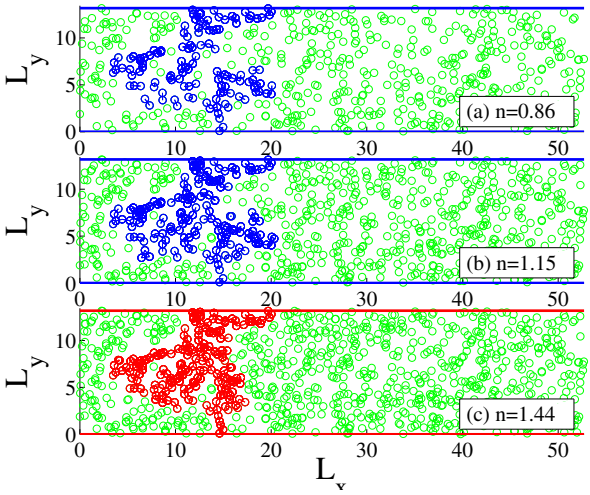


Fig. 1. The snapshots of evolution of two dimensional (2D) system as new circles representing tree crowns or bush are added and overall number density increases n. Two tree crowns or bush lie in the same cluster if the representing circles intersect. At lower densities (a) $n = 0.86$ and (b) $n = 1.15$ there is no percolating cluster in y direction. Circles that will form percolating cluster are blue while others are green. As density increases percolating cluster (red) is created, cf. (c) $n = 1.44$.

2 Percolation Probability Density

Monte Carlo simulations, coupled with an efficient cluster analysis algorithm and implemented on grid platform, are used to investigate the fire percolation probability dependence on density of a vegetation phase, and the surface geometry (shape) [22–25]. We consider two dimensional (2D) systems with isotropically placed circles representing tree crowns or bushes. Circles have all same unit radius and are randomly positioned and oriented inside the rectangular field of width L_x and height L_y. Two tree crowns or bushes lie in the same cluster if the representing circles intersect. System percolates if two opposite boundaries are connected with the same cluster, see Figure 1. The aspect ratio r is defined as the length of the rectangular system in percolating direction divided with the length in perpendicular direction. We define the normalized system size as a square root of the rectangular area $L = \sqrt{L_x L_y}$ (geometric average), which represent the length of the square system with the same area. The percolation behavior, i.e., the probability that will be able to find path to propagate, is studied in terms of the vegetation density $n = N/L^2$ where N is total number of trees/bushes.

From Fig. 2, one can see that average percolation density $\langle n \rangle_{L,r}$ for aspect ratio higher than one is monotonically decreasing function of the system size L. Somewhat surprising, for aspect ratios lower than one, $\langle n \rangle_{L,r}$ is not monotonic function and has local minimum. For small systems $\langle n \rangle_{L,r}$ is a decreasing function, which passes through n_c, reaches a minimum and after that converges to n_c from below. From general scaling arguments one would expect that for all finite

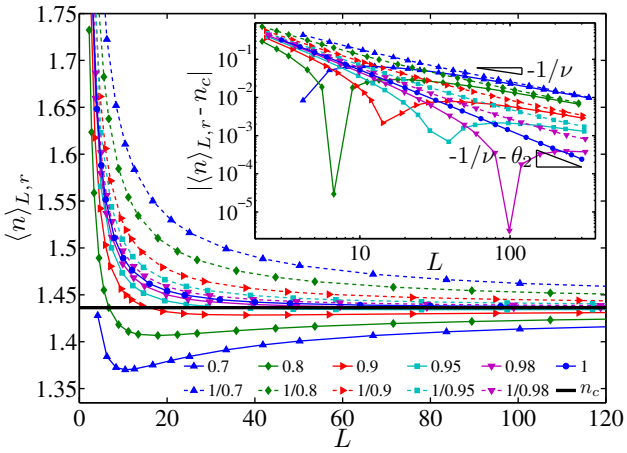


Fig. 2. The average percolation density $\langle n \rangle_{L,r}$ dependence on the system size L and aspect ratio r. The values are obtained from Monte Carlo simulations and calculated using Eq. (1). The values are given for aspect ratios $r = 0.7, 0.8, 0.9, 0.95, 0.98, 1$ (solid lines) and their inverse values $r = 1/0.7, 1/0.8, 1/0.9, 1/0.95, 1/0.98$ (dashed lines). The bold line denotes the expected values for the percolation threshold n_c. Inset: The same data is shown in logarithmic scale to demonstrate the same power law convergence of the r and $1/r$ pairs.

systems their convergence is governed by an exponent $-1/\nu$. For two-dimensional (2D) systems $\nu = 4/3$ [26].

Further it was showed that for lattice percolation on the square system leading exponent of the average concentration at which percolation first occurs is $-1/\nu - \theta$, where $\theta \approx 0.9$ [27]. These studies were performed for symmetric systems. In inset of Fig. 2, one can see that for large system sizes all the curves show power law convergence to percolation threshold n_c with exponent $-1/\nu$, except in the symmetric case, i.e., $r = 1$, where exponent is $-1/\nu - \theta_1$. Absolute values of the leading-order prefactors are the same for aspect ratios r and $1/r$.

The scaling behavior of the $\langle n \rangle_{L,r}$ can be described with generalized moment scaling function with aspect ratio dependent coefficients

$$\langle n \rangle_{L,r} = n_c + L^{-1/\nu} \sum_{i=1}^{\infty} a_i(r) L^{-\theta_i}, \tag{1}$$

where θ_i are generalized corrections to scaling exponents [31]. The coefficients of the first two order terms have form $a_1(r) \approx a_{1,1} \ln(r) + a_{1,2} \ln^3(r)$ and $a_2(r) \approx a_{2,1} + a_{2,2} \ln^2(r)$ (for detailed derivation of expression see Ref. [13]). The previous expansion implies a faster then $L^{-1/\nu}$ convergence of average percolation density $\langle n \rangle_{L,r}$ of symmetric system to its infinite-system value, characterized by an exponents θ_1. The standard deviation $\Delta_{L,r}$ can be described with expansion

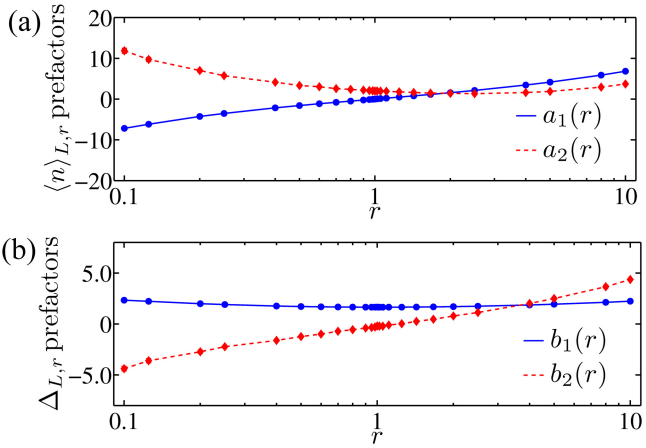


Fig. 3. Prefactors are shown for the two leading-order terms of generalized scaling function for average percolation density $\langle n \rangle_{L,r}$ (upper graph) and standard deviation $\Delta_{L,r}$ (lower graph). The first order prefactor of average percolation density is odd function on logarithmic scale, i.e., $a_1(r) = -a_1(1/r)$ and the first order prefactor of standard deviation is even function, i.e., $b_1(r) = b_1(1/r)$.

$$\Delta_{L,r} = L^{-1/\nu} \sum_{i=1}^{\infty} b_i(r) L^{-\theta_i}. \tag{2}$$

As one would expect, standard deviation is monotonically decreasing function of the system size L. The coefficients of the first two order terms have form $b_1(r) \approx b_{1,1} + b_{1,2}\ln^2(r)$ and $b_2(r) \approx b_{2,1}\ln(r) + b_{2,2}\ln^3(r)$, cf. Ref. [13].

From Monte Carlo simulation data we have obtained the first and second order terms of $\langle n \rangle_{L,r}$ in Eq. (1) by interpolation. Results of the analysis are shown in Fig. 3(a) and coefficients are given in Table 1. The influence of higher order terms were comparable or smaller then simulation data error and we could not extract them with sufficient precision. We obtain that the first order correction θ_1 is equal zero, as predicted by general scaling arguments in Ref. [26]. For the second order correction, we obtain $\theta_2 = 0.83(2)$ for $r = 1$. The residual aspect ratio dependence of θ_2 cannot be further analyzed without provision of retaining the first two terms in Eq. (1). The variance prefactors for two leading terms are shown in Fig. 3(b). The fitting coefficients $b_{i,j}$ are given in Table 1.

To illustrate impact of previous conclusions, we calculate safe (maximal) vegetation density n_{max} in fire protection strips. The results are given in Figure 4 for percolation probability tolerances $p = 15.73\%$ and 0.01%, i.e., probabilities that fire will find a passage. The curves are calculated using average value and standard deviation given in Eq. (1) and (2). Probability density function is approximated with normal distribution. It is helpful to understand that the maximal vegetation density depends on logarithm of strip length. This means that even for $L_x = 100L_y$ there is a high tolerance for vegetation density $n_{max} \approx 0.45$ for which probability of fire finding path through the strip is less then 0.01%.

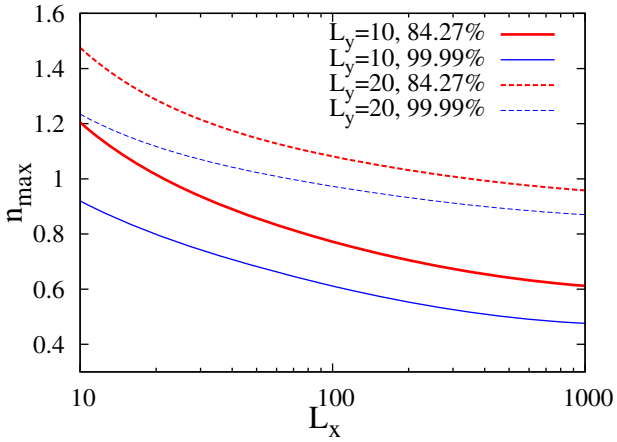


Fig. 4. Safe (maximal) vegetation density n_{max} in fire protection strips for which percolation probability is $p = 15.73\%$ (red) and 0.01% (blue). The curves are calculated using average value and standard deviation given in Eq. (1) and (2).

Table 1. Results for coefficients $a_{i,j}$ and $b_{i,j}$, where $i, j \in 0, 1$. The results are obtained using the least-square method.

	1,1	1,2	2,1	2,2
$a_{i,j}$	2.5(1)	0.11(3)	2.2(7)	1.0(3)
$b_{i,j}$	1.61(3)	0.14(1)	1.35(8)	0.11(2)

3 Fire Spreading on Percolating Networks

In this section we further extend percolation model to include combustion effects, where the whole forest/bush (i.e., vegetation and gases) is described with an equivalent point values at positions of its constituting parts. The assumptions for introducing this simple model are the following: The forest at the macroscopic scale can be considered as a random medium, with density n. This random medium, called vegetal phase, is composed of fuel (e.g., trees, trunks, bushes, etc.). One can consider that one of the main process in forest fire is heat transfer by radiation and convection. We consider that heat transfer is only possible between neighboring trees/bushes, all of the same unit radius. The main effects of the heat transfer are drying of the vegetation and vegetation pyrolysis that produce heat. We assume that hydrodynamics of gas which allows to bring the oxygen necessary to the combustion is fast and homogeneous. The energy released in pyrolysis leads to forest fire propagation.

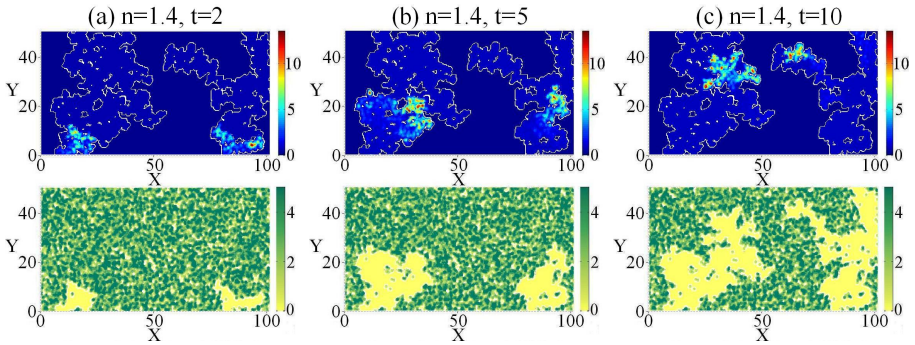


Fig. 5. Snapshots visualizing the propagation of fire through vegetal phase at $t = 2, 5$, and 10. System has initial vegetal phase density $n = 1.4$ and fire is initiated at its lower boundary. Upper plot represents local temperature distribution and lower local vegetal phase density.

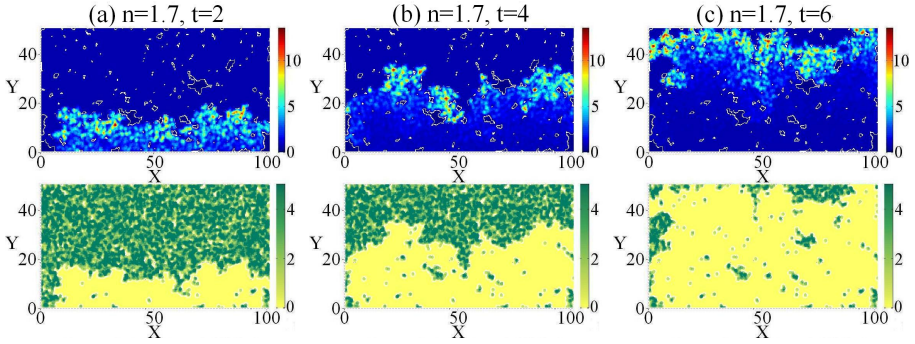


Fig. 6. Snapshots visualizing the propagation of fire through vegetal phase at $t = 2, 4$, and 6. System has initial vegetal phase density $n = 1.7$ and fire is initiated at its lower boundary. Upper plot represents local temperature distribution and lower local vegetal phase density.

We have simplified the general model introduced elsewhere, cf. Ref. [28–30], in order to obtain the following system of equations. The balance of energy or thermal equation is

$$\frac{\partial T_{\mathrm{f}}^{j}}{\partial t} = \lambda_{\mathrm{f}} \triangle T_{\mathrm{f}}^{j} + R^{j} + \lambda_{\mathrm{i}}(T_{\mathrm{f}}^{j} - T_{\mathrm{i}}), \tag{3}$$

where T_{f} is temperature of vegetal phase at mass point j, λ_{f} equivalent heat conductivity between vegetal phase, λ_{i} equivalent heat conductivity to environment and T_{i} environment temperature. We assume for simplicity parabolic temperature distribution between neighboring points. The pyrolytic heat source is given by $R^{j} = (k(T_{\mathrm{f}}^{j})/C_p)Y_{\mathrm{f}}^{j}$ where Y_{f}^{j} is fraction of vegetation phase in mass point, C_p effective heat capacity and k is reaction rate. In the present model $Y_{\mathrm{f}}^{j} = 1$ is corresponding to the vegetation phase green at the mass point and $Y_{\mathrm{f}}^{j} = 0$ burned. Reaction rate $k(T)$ is defined by the standard Arrhenius expression,

i.e., $k(T) = Be^{-E_A/k_B T}$, where B is prefactor, k_B Boltzmann constant and E_A activation energy. The balance of mass at point j is written generically

$$\frac{\partial Y_f^j}{\partial t} = -k(T_f^j)Y_f^j. \tag{4}$$

Comparison of fire propagation between two simulations is given in Figures 5 and 6. The two simulations have identical inputs $E_A/k_B = 0.1$, $B = 1000$, $\lambda_i = 1.2$, $\lambda_f = 1.$ and $C_p = 10.$ but different spatial densities $n = 1.4$ and 1.7. The fire is initiated at its lower boundary. As one could expect, at lower density $n = 1.4$ not whole surface can be visited by the fire. Still, we observe that in both cases fire propagation is not homogeneous and it depends on local structure of the vegetation phase. We also observe a local auto-extinction events. The fire is spreading faster through the clusters of vegetation phase, and it slows down when vegetation becomes rate. Therefore differences in propagation are not large inside clusters; differences in variance in spread rate is only significant between vegetation clusters. Therefore, at higher densities where connections between vegetation clusters are abundant fire spread rate is less variable, cf. Figure 7.

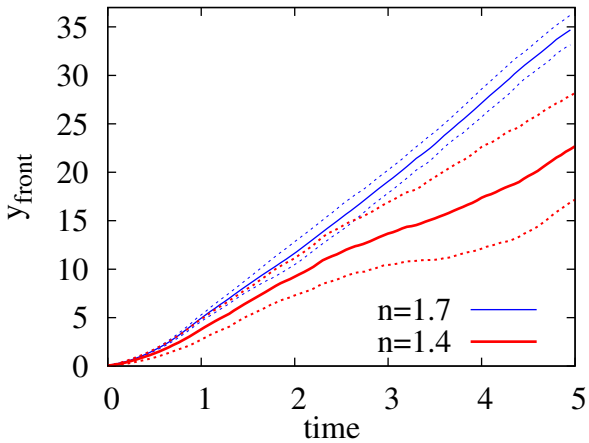


Fig. 7. Evolution of the fire front in time for two vegetation densities $n = 1.4$ and $n = 1.7$. Bold line represents mean value of fire front position y_{front} and dashed lines are delimiting 68% confidence band ($y_{front} \pm \sigma$).

4 Cascade Failures

In the present proceeding we adopt the model of cascading failures caused by overload presented in [2] to investigate the overload breakdown problem when edges (rather than nodes) are sensitive to overloading. Within this model, it is assumed that, at each time step, every node provides (receives) flow to (from) every other node of the network with an equal share and the flow is forwarded

along the shortest path. The edge's initial load L is defined by means of the betweenness centrality, which describes the number of all shortest paths through the edge. The capacity C_i of edge i, which defines the maximum load that the edge can handle at each time step, is set to be proportional to the edge initial load $C_i = (1+\alpha)L_i$, where the constant $\alpha \geq 0$ is a tolerance parameter. An edge is overloaded and fails if $L_i > C_i$. When for any reason an edge fails, the traffic which used to go through this edge is redistributed to a new shortest path. This results in a network-wide redistribution of traffic load. As a consequence of this redistribution, some edges have to carry a larger load than before. If this new load exceeds the capacity of these edges, then the respective edges will also fail, triggering a new load redistribution with possible, subsequent overload failures of other edges. This eventually leads to a cascade of failures, after which only a fraction of the nodes is still functioning. This fraction does not necessarily form a connected network. The largest of these subnetworks, i.e., the one containing the largest number of nodes, is called the giant component G.

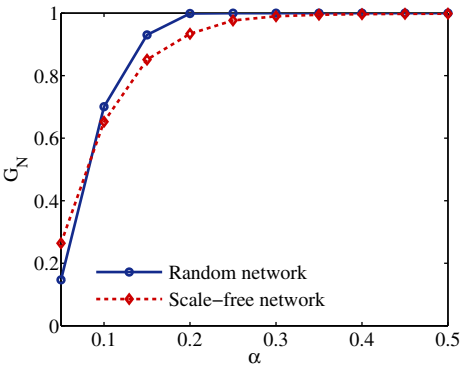


Fig. 8. The average giant component G_N vs tolerance parameter α

Here we consider the cascading failures triggered by removing a single edge. To measure the network functionality we consider the size of the giant component G. We illustrate how this model works in practice by considering random network and scale-free network. Random network can be generated by the Erdős-Rényi model [11], where each pair of nodes is linked with probability p. To generate scale-free network we use standard Barabàsi-Albert model [12]. In both cases we have constructed networks with N=1000 nodes. For random network, edge probalility has been set to $p = 0.005$. In scale-free model, starting from $N_0 = 3$ nodes, one node with $m = 2$ edges is attached iteratively.

To measure the robustness degree of the whole networks against cascading failures, we remove every edge in a network one by one and calculate the corresponding results, e.g., removing edge i and calculating G_i after the cascading process is over. To quantify the robustness of the whole network, we adopt the average giant component, i.e., $G_N = \frac{1}{|E|}\sum_{i\in E} G_i$. The set of edges in the network is denoted with E. The results in function of the tolerance parameter α

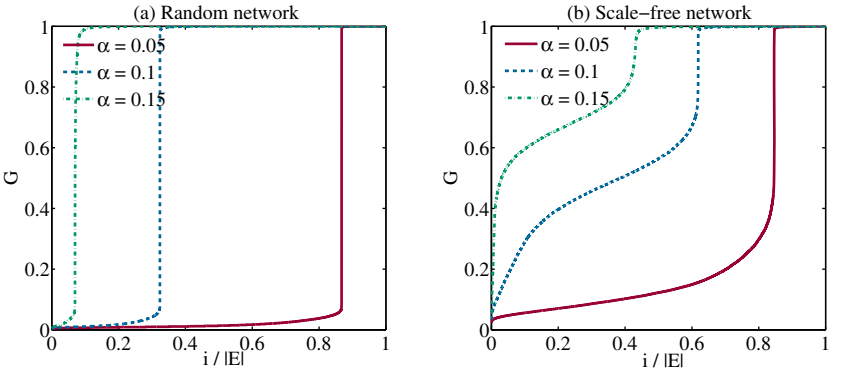


Fig. 9. The giant component size distribution after the system has relaxed to a stationary state for (a) random network and (b) scale-free network

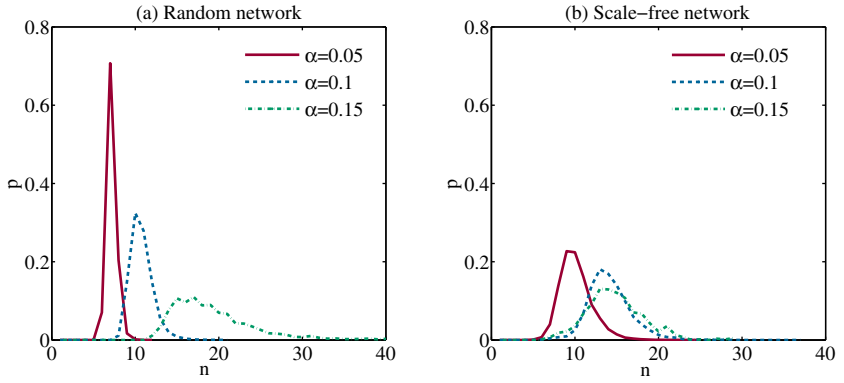


Fig. 10. Probability distributions for the total number of step n of the cascade failure for (a) random network and (b) scale-free network

are summarized in Fig. 8. Clearly, a higher value of α results in a larger giant component after cascade failures finished. One can also observe that for small tolerance, i.e., $\alpha < 0.1$, the random network is more sensitive to cascade failures. This changes for larger tolerances, and scale-free network becomes more sensitive due to more heterogeneous distribution of node degrees and capacities in the scale-free network. Fig. 9 plots size of the giant component from all the possible scenarios of removing a single edge in one network. Simulation results for random network show that giant component discontinuously changes value. We can observe that in random network for $\alpha = 0.05$, removal of a single edge will lead to the collapse of the system under overload failures in more than 80% cases. For $\alpha = 0.1$, percentage of critical edges in network is higher than 30%, and for $\alpha = 0.15$ it is about 10%. For the scale-free network, for the same values of α, we obtain continuous change.

To get another insight into the mechanism of the breakdown, we consider histograms of the total number of step n of the cascade failure in which the size of the giant component is reduced by more than 50%, Fig 10. The average number of cascading failure steps n increases with tolerance parameter α. It is interesting to note that both random and scale-free networks will disintegrate in roughly the same time. The main difference is width of the distribution. In case of random network the distribution is very sharp and more then 70% of networks will disintegrate after 8 steps when $\alpha = 0.05$.

5 Conclusion

In this proceeding we demonstrate how theory of complex systems and the statistical physics of networks may provide us with methods for disaster propagation prediction. These methods allow one to gain a better understanding of the dynamics of disaster spreading and to derive results indicating how to fight them best. We have specifically presented three combinations of networks and specific disaster processes on them. These processes interact with given network structure in different ways yet have a generic thread between them - a variability in outcome due to the local properties of the network structure.

As the main parameters, we have considered the overall density of vegetal phase in fire spreading models and tolerance of the edge to increase of the load. By means of simulations and theoretical model, we have examined probability that fire will propagate through fire protection strip. We have also compared uncertainty in prediction of fire propagation rate due to the local inhomogeneities of the vegetation cover. Important conclusion is that uncertainty in fire propagation rate is especially high for less dense vegetation cover and close to fire percolation point. We have also measured uncertainty in cascade failures of network infrastructure. A model where edges have limited capacity is studied. The behavior of random-network was found to be ambiguous. In comparison to scale-free network, with high and medium values of the tolerance parameter (i.e., $\alpha > 0.1$) most the network is prone to failure of the most edges. On the other hand, failure of one of the critical edges, the network will disintegrate completely.

Acknowledgements. The authors acknowledge support by the Ministry of Science of the Republic of Serbia, under project No. ON171017. Numerical simulations were run on the AEGIS e-Infrastructure, supported in part by FP7 projects EGIInSPIRE, PRACE-1IP, PRACE-2IP, and HP-SEE. The authors also acknowledge support received through SCOPES grant IZ73Z0-128169 of the Swiss National Science Foundation.

References

1. Contreras, M.A., Parsons, R.A., Chung, W.: Forest Ecol. Manage. 264, 134 (2012)
2. Motter, A.E., Lai, Y.-C.: Phys. Rev. E 66, 065102 (2002)

3. Smiljanic, J., Stankovic, I.: Physica A, doi:10.1016/j.physa.2013.01.033
4. Newman, E.J., Forrest, S., Balthrop, J.: Phys. Rev. E 66, 035101 (2002)
5. Kalapala, V., Sanwalani, V., Clauset, A., Moore, C.: Phys. Rev. E 73, 026130 (2006)
6. Buzna, L., Peters, K., Ammoser, H., Kühnert, C., Helbing, D.: Phys. Rev. E 75, 056107 (2007)
7. Morandini, F., Silvani, X., Rossi, L., Santoni, P.-A., Simeoni, A., Balbi, J.-H., Rossi, J.L., Marcelli, T.: Fire Safety J. 41, 229 (2006)
8. Bak, P., Tang, C., Wiesenfeld, K.: Phys. Rev. A 38, 364 (1988)
9. Clar, S., Drossel, B., Schwabl, F.: J. Phys. Condens. Matter 8, 6803 (1996)
10. Zinck, R.D., Grimm, V.: Am. Nat. 174, 170 (2009)
11. Bollobás, B.: Random Graphs. Academic, London (1985)
12. Barabási, A.-L., Albert, R.: Science 286, 509 (1999)
13. Zezelj, M., Stankovic, I., Belic, A.: Phys. Rev. E 85, 021101 (2012)
14. Linn, R.R., Winterkamp, J.L., Weise, D.R., Edminster, C.: Int. J. Wild. Fire 19, 179 (2010)
15. Albert, R., Albert, I., Nakarado, G.L.: Phys. Rev. E 69, 025103 (2004)
16. Crucciti, P., Latora, V., Marchiori, M.: Phys. Rev. E 69, 045104 (2004)
17. Motter, A.E.: Phys. Rev. Lett. 93, 098701 (2004)
18. Schäfer, M., Scholz, J., Greiner, M.: Phys. Rev. Lett. 96, 108701 (2006)
19. Buldyrev, S.V., Parshani, R., Paul, G., Stanley, H.E., Havlin, S.: Nature 464, 08932 (2010)
20. Parshani, R., Buldyrev, S.V., Havlin, S.: PNAS 108, 1007 (2011)
21. Simonsen, I., Buzna, L., Peters, K., Bornholdt, S., Helbing, D.: Phys. Rev. E 100, 218701 (2008)
22. Newman, M.E.J., Ziff, R.M.: Phys. Rev. E 64, 016706 (2001)
23. Stanković, I.E., Kröger, M., Hess, S.: Comp. Physics Comm. 145, 371 (2002)
24. Balaž, A., Prnjat, O., Vudragović, D., Slavnić, V., Liabotis, I., Atanassov, E., Jakimovski, B., Savić, M.: J. Grid. Comput. 9, 135 (2011)
25. Li, J., Zhang, S.-L.: Phys. Rev. E 80, 040104 (2009)
26. Stauffer, D., Aharony, A.: Introduction to Percolation Theory, 2nd revised edn. Taylor and Francis, London (2003)
27. Ziff, R.M., Newman, M.E.J.: Phys. Rev. E 66, 016129 (2002)
28. Margerit, J., Séro-Guillaume, O.: Int. J. Heat Mass Transf. 45, 1723 (2002)
29. Kuittu, M.-P., Haataja, M., Provatas, N., Ala-Nissila, T.: Phys. Rev. E 58, 1514 (1998)
30. Méndez, V., Llebot, J.E.: Phys. Rev. E 56, 6557 (1997)
31. Hovi, J.-P., Aharony, A.: Phys. Rev. E 53, 235 (1996)

Determination of Zone of Flow Instability in a Gas Flow Past a Square Particle in a Narrow Microchannel

Kiril Shterev and Stefan Stefanov

Institute of Mechanics, Bulgarian Academy of Sciences,
Acad. G. Bonchev Str., Block 4, Sofia 1113, Bulgaria
kshterev@imbm.bas.bg

Abstract. The rapidly emerging industry of micro-electro-mechanical devices gives rise to new potential microfluidic applications. The analysis of the possible flow regimes is an important task of any microfluidic investigation. For a gas flow the transition between steady and unsteady regimes occurs at small Knudsen number $Kn < 0.1$ ($Kn = \ell_0/L$, where ℓ_0 is the mean free path of the gas molecules and L is the characteristic length). A continuum approach based on the Navier-Stokes-Fourier equations is applicable for this investigation. On the other side, the microfluidic application requires the problem to be investigated starting at very low Mach numbers ($M = 0.1$), close to incompressible regime. This makes pressure based methods very suitable for this investigation. The system of Navier-Stokes-Fourier equations is calculated numerically using pressure based algorithm SIMPLE-TS 2D. The results are validated by comparing them to data obtained by using molecular approach (direct simulation Monte Carlo (DSMC) method).

1 Introduction

All devices with character dimensions between $1\mu m$ and $1mm$ are called microdevices. Micro mechanical devices are rapidly emerging technologies, where new potential applications are continuously being developed and fluid flow regime is important for their design [1], [2]. In this paper is investigated the established regime of a flow past square in a microchannel as a function of rarefaction (Knudsen number) and velocity (Mach number).

Usually, a microchannel gas flow is modelled by using either continuum approach or rarefied gas dynamics methods. The Knudsen number (Kn),defined as a ratio of the mean free path ℓ_0 to the macroscopic length scale of a physical system L,

$$Kn = \frac{\ell_0}{L} \tag{1}$$

is a non-dimensional parameter that determines the degree of gas rarefaction, and respectively, the degree of appropriateness of the continuum model. According Knudsen number can be defined the regimes [3]:

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_5, © Springer International Publishing Switzerland 2014

- *continuum flow*, for $Kn < 0.01$. Non-slip boundary conditions between the surfaces of bodies and fluid are applicable. The flow is described by the Navier-Stokes equations. In [4] is proposed this regime to be valid for $Kn < 0.001$.
- *slip-flow regime*, for $0.01 \leq Kn < 0.1$. The Navier-Stokes equations can be applied to the flow, but at the surface, the gas "slips": these last molecular layers have a velocity different from that of the surface.
- *transition regime*, for $0.1 \leq Kn < 10$. The Navier-Stokes equations are not valid.
- *free-molecule regime*, for $Kn \geq 10$. The gas must be described by the Boltzmann equation.

For larger Knudsen numbers, where the non-equilibrium effects are significant, a molecular approach, based on kinetic theory models [5], [6], or the particle DSMC (Direct Simulation Monte Carlo) method [7] has to be used. In the present paper, we restrict ourself to the use of Navier-Stokes-Fourier continuum model [8] with state-depended transport coefficients determined by the first approximation of the Chapman-Enskog theory for low Knudsen numbers. In our opinion, this model captures the basic flow effects of the motion of compressible viscous heat-conducting gas in continuum limit.

The peculiarity of fluid flows in and around micro-electro-mechanical-systems (MEMS) is the low speed fluid. In some cases the Mach number can be close to zero (incompressible limit). The pressure based methods are used to calculate incompressible fluid flows. This makes pressure based methods very suitable to be used here.

Investigations of final state of incompressible fluid past a square with non-slip boundary conditions are placed in many papers, see [9]. In this paper is investigated final state of flow as function of Knudsen and Mach numbers for a gas flow past square in a long, narrow microchannel. The investigated Mach numbers are 0.1, 0.2, 0.3 and 0.4, where local Mach number vary from 0.1 to ~ 0.8, see Fig. 2. To be able to calculate all problems with one algorithm is used iterative pressure based finite volume method (FVM) with improved convergence within a time step (SIMPLE-TS) compared to the other standard FVM iterative schemes. The algorithm SIMPLE-TS (Time Step) is presented in details in [10]. In this investigation is used parallel version of algorithm SIMPLE-TS. The parallel organization is presented in details in [11] and [12]. To reduce computational resources is used second order total variation diminishing (TVD) scheme with limiter SUPERBEE [13] for calculations of convective terms and density in middle points. The implementation, validation and mesh convergence of TVD SUPERBEE scheme, which is used in algorithm SIMPLE-TS is described in [14]. The problems are calculated on clusters of e-Infrastructure in South-Eastern Europe, see [15].

The molecular DSMC approach used here to validate the SIMPLE-TS for moderate Knudsen numbers was applied previously in series of papers, [8], [16], [17], for studying transition regimes and final state of 2D and 3D Rayleigh-Bénard flow of a rarefied gas as a function of Knudsen and Froude numbers.

2 Continuum Model Equations

A two dimensional system of equations describing the unsteady flow of viscous, compressible, heat conductive fluid can be expressed in a general form as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \tag{2}$$

$$\begin{aligned}\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u u)}{\partial x} + \frac{\partial(\rho v u)}{\partial y} = \rho g_x - A\frac{\partial p}{\partial x} + B\left[\frac{\partial}{\partial x}\left(\Gamma\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma\frac{\partial u}{\partial y}\right)\right] \\ + B\left\{\frac{\partial}{\partial x}\left(\Gamma\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma\frac{\partial v}{\partial x}\right) - \frac{2}{3}\frac{\partial}{\partial x}\left[\Gamma\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right]\right\}\end{aligned} \tag{3}$$

$$\begin{aligned}\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho u v)}{\partial x} + \frac{\partial(\rho v v)}{\partial y} = \rho g_y - A\frac{\partial p}{\partial y} + B\left[\frac{\partial}{\partial x}\left(\Gamma\frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma\frac{\partial v}{\partial y}\right)\right] \\ + B\left\{\frac{\partial}{\partial y}\left(\Gamma\frac{\partial v}{\partial y}\right) + \frac{\partial}{\partial x}\left(\Gamma\frac{\partial u}{\partial y}\right) - \frac{2}{3}\frac{\partial}{\partial y}\left[\Gamma\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right]\right\}\end{aligned} \tag{4}$$

$$\begin{aligned}\frac{\partial(\rho T)}{\partial t} + \frac{\partial(\rho u T)}{\partial x} + \frac{\partial(\rho v T)}{\partial y} = C^{T1}\left[\frac{\partial}{\partial x}\left(\Gamma^{\lambda}\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma^{\lambda}\frac{\partial T}{\partial y}\right)\right] \\ + C^{T2}.\Gamma.\Phi + C^{T3}\frac{Dp}{Dt}\end{aligned} \tag{5}$$

$$p = \rho T \tag{6}$$

where:

$$\Phi = 2\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right] + \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)^2 - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^2 \tag{7}$$

u is the horizontal component of velocity, v is the vertical component of velocity, p is pressure, T is temperature, ρ is density, t is time, x and y are coordinates of a Cartesian coordinate system. The reference quantities used to scale the system of equations (2) - (6) will be defined later. The parameters A, B, g_x, g_y, C^{T1}, C^{T2}, C^{T3} and diffusion coefficients Γ and Γ^{λ}, given in Eqs. (2)-(6), depend on the gas model and the equation non-dimensional form. The system of equations (2) - (6) is solved by using the algorithm SIMPLE-TS [10]. A second order central difference scheme is employed for the approximation of the diffusion terms and second order TVD SUPERBEE scheme to approximate convective terms and density in middle points.

The system of equations (2) - (6) is given in a general form of the Navier-Stokes-Fourier equations. For a gaseous microflow description we use the model of a compressible, viscous hard sphere gas with diffusion coefficients determined by the first approximation of the Chapman-Enskog theory for low Knudsen numbers [8]. For a hard-sphere gas, the viscosity coefficient μ and the heat conduction coefficient λ read (first approximations are sufficient for our considerations) as:

$$\mu = \mu_h\sqrt{T},\ \mu_h = (5/16)\rho_0 l_0 V_{th}\sqrt{\pi} \tag{8}$$

$$\lambda = \lambda_h \sqrt{T},\ \lambda_h = (15/32) c_p \rho_0 l_0 V_{th} \sqrt{\pi} \tag{9}$$

The Prandtl number is given by $Pr = 2/3$, $\gamma = c_p/c_v = 5/3$. The dimensionless system of equations (2) - (6) is scaled by the following reference quantities, as given in [8]: molecular thermal velocity $V_0 = V_{th} = \sqrt{2RT_0}$ for velocity, for length - square size a (Fig. 1), for time - $t_0 = a/V_0$, the reference pressure (p_0) is the pressure at the inflow of the channel, the reference temperature (T_0) is equal to the channel walls, reference density (ρ_0) is calculated using equation of state (6), the calculated case neglects the influence of gravity field, therefore $g_x = g_y = 0$. The corresponding non-dimensional parameters in the equation system (2) - (6) read as follows:

$$\begin{gathered} A = 0.5,\ B = \frac{5\sqrt{\pi}}{16} Kn,\ \Gamma = \Gamma^{\lambda} = \sqrt{T}, \\ C^{T1} = \frac{15\sqrt{\pi}}{32} Kn,\ C^{T2} = \frac{\sqrt{\pi}}{4} Kn,\ C^{T3} = \frac{2}{5} \end{gathered} \tag{10}$$

The algorithm SIMPLE-TS for calculation of gas microflows is validated in [10] and [14].

3 Problem Formulation, Results and Discussion

The final state flow as a function of Knudsen and Mach numbers is determined for a square particle with size a confined in a plane microchannel (height H_{ch}) as shown in Fig. 1. The blockage ratio $B = a/H_{ch}$ is equal to $B = 3$, the inflow length is L_a. The problem is considered in a local Cartesian coordinate system, which is moving with the particle. Thus for an observer moving along with the particle the problem is transformed to a consideration of a gas flow past a stationary square confined in a microchannel with moving walls. Velocity-slip

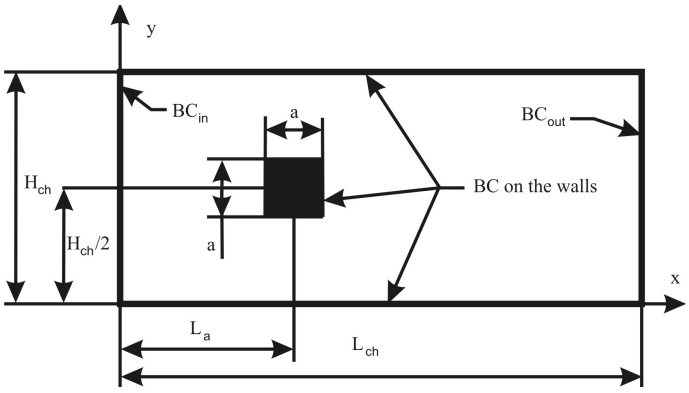


Fig. 1. Flow geometry for a square-shaped particle with size a confined in a channel with length L_{ch} and height H_{ch}

and temperature-jump boundary conditions are imposed on the walls of the channel and the square. The velocity-slip BC is given as: $v_s - v_w = \zeta \left.\frac{\partial v}{\partial n}\right|_s$, where v_s is velocity of the gas at the solid wall surface, v_w is velocity of the wall, $\zeta = 1.1466.Kn_{local} = 1.1466.Kn/\rho_{local}$, Kn_{local} is the local Knudsen number, ρ_{local} is the local density, $\left.\frac{\partial v}{\partial n}\right|_s$ is the derivative of velocity normal to the wall surface. The temperature-jump boundary condition is: $T_s - T_w = \tau \left.\frac{\partial T}{\partial n}\right|_s$, where T_s is temperature of the gas at the wall surface , T_w is temperature of the wall, $\tau = 2.1904.Kn_{local} = 2.1904.Kn/\rho_{local}$, $\left.\frac{\partial T}{\partial n}\right|_s$ is the derivative of temperature normal to the wall surface.

In this paper we are interested in the final flow state established after passing the transient period of flow evolution. The flow fields of horizontal velocity and temperature are shown in Fig. 2 and Fig. 3 for Knudsen numbers $Kn = 0.0105$ and 0.0100, respectively. The fields of a steady state flow (for example at $Kn = 0.0105$) is symmetric, while behind the square the final state fields of an unsteady state flow (for example at $Kn = 0.0100$) are constantly changing by creation of non-symmetric patterns in well-known form of Karman vortex street.

The final states (for both steady and unsteady flow) are investigated for Mach numbers $M = 0.1, 0.2, 0.3$ and 0.4. To this aim series of numerical calculation varying Knudsen number are performed for each considered Mach number. Numerically, the final state is considered to have been reached when the regime situation is established and the influence of inflow and outflow boundaries becomes negligible. The implementation of inflow and outflow boundary condition without fixing pressure in computational domain and at the same time keeping mass conservation is a task , which is not achieved yet completely (as seen from the published literature). This problem is mentioned in [18]. Thus, in our considerations the pressure at the inflow boundary is fixed. The influence of this boundary condition can be neglected, when disturbances of moving square do not reach the channel inflow. To neglect influence of inflow boundary conditions for $M = 0.1$ one has to use $L_a = 14000$, Fig. 4. a), i.e. the microchannel becomes extremely long and requires a lot of computational resources. The obtained results are compared to 10 times shorter inflow channel length $L_a = 1400$, Fig. 4. a). The pressure difference in stagnation point is about $\sim 1\%$ (Fig. 4. a)) and there is no difference in the transition point from steady to unsteady regime for both lengths. Respectively, for $M = 0.4$ (Fig. 4. b)) the pressure difference in stagnation point is $\sim 12\%$ and the difference of the determined Knudsen number in the transition point is $\sim 20\%$. Thus, to ensure a correct determination of the transition from steady to unsteady flow one must use sufficiently extended computational domain in front of the square for cases, when Mach number is higher then $M = 0.1$.

From the results obtained for the cases calculated at Mach and Knudsen numbers shown in Fig. 5, one can observe clear functional close to linear dependency between Knudsen and Mach critical numbers representing the neutral line separating the steady state flow zone from the unsteady state one.

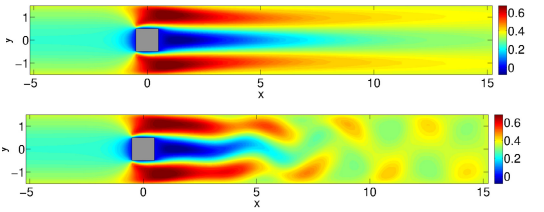


Fig. 2. Horizontal velocity field for $M = 0.4$ and $Kn = 0.0105$ final state - steady flow (upper part) and $M = 0.4$ and $Kn = 0.0100$ final state - unsteady flow (lower part)

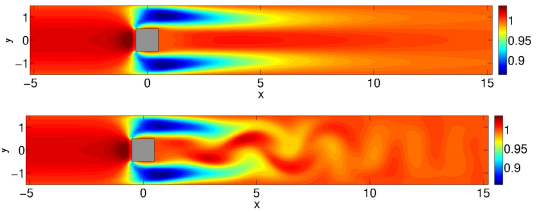


Fig. 3. Temperature field for $M = 0.4$ and $Kn = 0.0105$ final state - steady flow (upper part) and $M = 0.4$ and $Kn = 0.0100$ final state - unsteady flow (lower part)

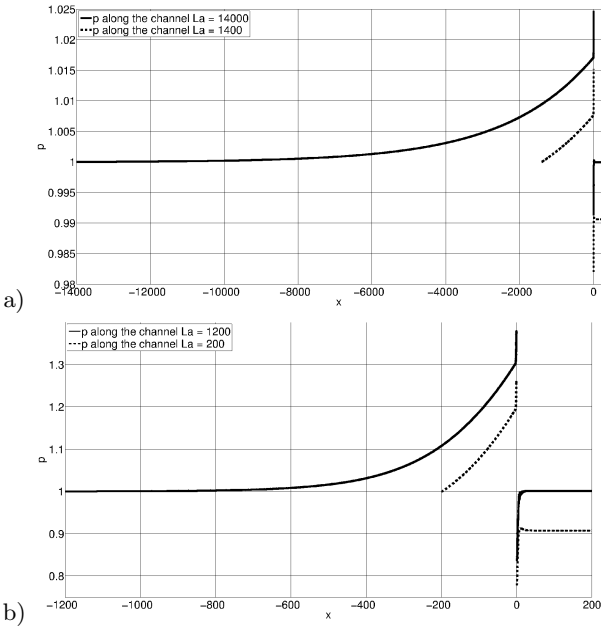


Fig. 4. Pressure along the channel central line for a) $M = 0.1$, $Kn = 0.00315$ and compared cases for length of computational domain in front of the square $La = 14000$ and $La = 1400$ and b) $M = 0.4$, $Kn = 0.01$ and compared cases for $La = 1200$ and $La = 200$

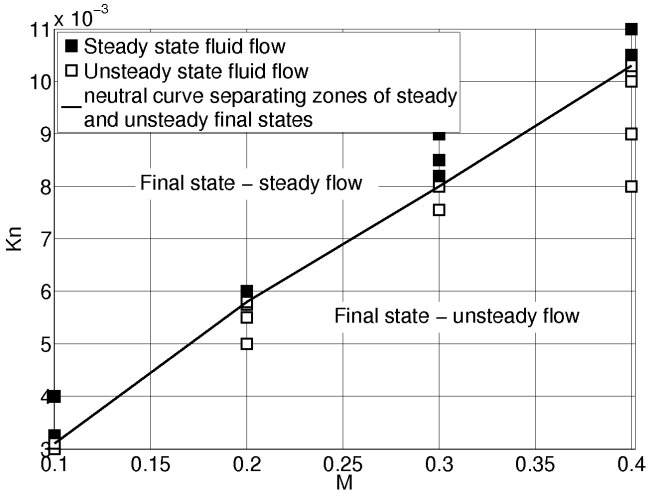


Fig. 5. Final state of flow as function of Knudsen and Mach numbers. The solid line presents the neutral curve separating zones of steady and unsteady final states.

4 Conclusions

A flow past square in a narrow microchannel for Mach numbers 0.1, 0.2, 0.3 and 0.4 is considered for various small Knudsen numbers. As a result the final flow state is determined as function of Mach and Knudsen numbers. The results show that the transition between steady and unsteady flow regime obeys a functional dependency between critical Mach and Knudsen numbers, which is close to linear.

Acknowledgments. The first author appreciates the financial support provided by the NSF of Bulgaria under Grant No DMU 03/37, since 2011. This work makes use of results produced by the High-Performance Computing Infrastructure for South East Europe's Research Communities (HP-SEE), a project co-funded by the European Commission (under contract number 261499) through the Seventh Framework Programme. HP-SEE involves and addresses specific needs of a number of new multi-disciplinary international scientific communities (computational physics, computational chemistry, life sciences, etc.) and thus stimulates the use and expansion of the emerging new regional HPC infrastructure and its services. Full information is available at: http://www.hp-see.eu/.

References

1. Colin, S.: Rarefaction and compressibility effects on steady and transient gas flows in microchannels. Microfluidics and Nanofluidics 1, 268–279 (2005)

2. Nabavi, M.: Steady and unsteady flow analysis in microdiffusers and micropumps: a critical review. Microfluidics and Nanofluidics 7, 599–619 (2009)
3. Schaaf, S.A., Chambre, P.L.: Flow of Rarefied Gases. Princeton University Press, USA (1961)
4. el Hak, M.G.: The fluid mechanics of microdevices: The freeman scholar lecture. ASME J. Fluid Eng. 121, 5–33 (1999)
5. Cercignani, C.: Rarefied Gas Dynamics. From Basic Concept to Actual Calculations. University Press, Cambridge (2000)
6. Sone, Y.: Molecular Gas Dynamics: Theory, Techniques, and Applications. Birkhäser, Boston (2007)
7. Bird, G.A.: Molecular, Gas Dynamics and the Direct Simulation of Gas Flows. Clarendon Press, Oxford (1994)
8. Stefanov, S., Roussinov, V., Cercignani, C.: Rayleigh-Bénard flow of a rarefied gas and its attractors. I. Convection regime. Physics of Fluids 14(7), 2255–2269 (2002)
9. Sohankar, A., Davidson, L., Norbeg, C.: Numerical Simulation of Unsteady Flow Around a Square Two-Dimensional Cylinder, pp. 517–520 (1995)
10. Shterev, K.S., Stefanov, S.K.: Pressure based finite volume method for calculation of compressible viscous gas flows. Journal of Computational Physics 229(2), 461–480 (2010)
11. Shterev, K.S., Stefanov, S.K.: A parallelization of finite volume method for calculation of gas microflows by domain decomposition methods. In: Lirkov, I., Margenov, S., Waśniewski, J. (eds.) LSSC 2009. LNCS, vol. 5910, pp. 523–530. Springer, Heidelberg (2010)
12. Shterev, K.S., Stefanov, S.K., Atanassov, E.I.: A parallel algorithm with improved performance of finite volume method (simple-ts). In: Lirkov, I., Margenov, S., Waśniewski, J. (eds.) LSSC 2011. LNCS, vol. 7116, pp. 351–358. Springer, Heidelberg (2012)
13. Roe, P.L.: Some contributions to the modelling of discontinuous flows. Lectures in Applied Mathematics 22, 163–193 (1985)
14. Shterev, K.S., Ivanovska, S.: Comparison of some approximation schemes for convective terms for solving gas flow past a square in a microchannel. In: AIP Conference Proceedings, vol. 1487(1), pp. 79–87 (2012)
15. Balaž, A., Prnjat, O., Vudragović, D., Slavnić, V., Liabotis, I., Atanassov, E., Jakimovski, B., Savić, M.: Development of Grid e-Infrastructure in South-Eastern Europe. Journal of Grid Computing 9(2), 135–154 (2011)
16. Stefanov, S., Roussinov, V., Cercignani, C.: Rayleigh-Bénard flow of a rarefied gas and its attractors. II. Chaotic and periodic convective regimes. Physics of Fluids 14(7), 2270–2288 (2002)
17. Stefanov, S., Roussinov, V., Cercignani, C.: Rayleigh-Bénard flow of a rarefied gas and its attractors. III. Three-dimensional computer simulations. Physics of Fluids 19(12), 124101 (2007)
18. GadelHak, M.: MEMS: Introduction and Fundamentals. CRC Press (2005)

Quenched Hadron Spectroscopy Using FermiQCD

Rudina Zeqirllari[1], Dafina Xhako[2], and Artan Boriçi[1]

[1] Departament of Physics, Faculty of Natural Sciences,
University of Tirana
Blv. Zog I, Tiranë, Albania
rudinazeqirllari@gmail.com
[2] Department of Engineering Science,
Professional Studies Faculty
"Alexander Moisiu" University, Durres, Albania

Abstract. FermiQCD is a C++ library for fast development of parallel Lattice Quantum Field Theory computations. It has been developed following a top - down fully Object Oriented design approach with focus on simplicity of use. We present simulations for the hadron spectrum with the Wilson action in quenched QCD carried out on the BG HPC cluster, using this tool kit. Testing FermiQCD as a tool kit for parallel lattice QCD applications and see how parallel calculations are implemented in this package are some aims of this work. Simulations are made with the plaquette gauge action on 84, 124 and 164 lattices at three lattices spacing, for a total number of 300 SU(3) gauge configurations. The masses of pi and rho – meson, nucleon and delta baryons are computed for these conditions. For a constant physical volume we study the dependence of the hadrons masses on the lattice spacing. After chiral and continuum extrapolations, the agreement of the calculated mass spectrum for this kind of fermions with experiment is reasonable. Our application seems to have a good scalability on parallel computers.

Keywords: Lattice QCD, FermiQCD, hadron spectrum, Wilson fermions.

1 Introduction

1.1 Introduction to Lattice QCD

Quantum Chromodynamics (QCD) is the theory of strong interaction. It describes how quarks and gluons interact via the strong force and predicts in principle the masses of their bound states, known as hadrons. This kind of predictions are beyond the reach of the traditional perturbative methods of quantum field theory and can only be done by the numerical simulation of QCD on a lattice of space-time points ("Lattice QCD"). Substantial computing resources must be applied to these simulations.

In lattice QCD, fields representing quarks are defined at lattice sites, while the gluon fields are defined on the links connecting neighboring sites.

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/ 978-3-319-01520-0_6, © Springer International Publishing Switzerland 2014

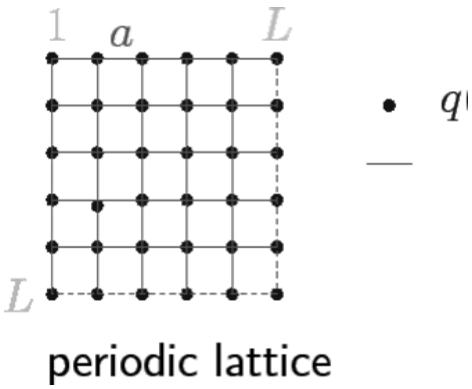


Fig. 1. Non - interacting quarks on a lattice. The quarks are representing by the sites, while gluon fields are defined on the links connecting neighboring sites.

This approximation approaches continuum QCD as the spacing between lattice sites is reduced to zero. [1]. Because the computational cost of numerical simulations can increase as the lattice spacing decreases, often the results are extrapolated to a = 0 by repeated calculations at different lattices spacing a. Numerical lattice QCD calculations using Monte Carlo methods can be extremely computationally intensive, requiring the use of the largest available supercomputers.

Another problem in LQCD is the fermion doubling. It is a problem that appear when naively we try to put fermions fields on a lattice. It consists in the appearance of other states, such that one ends up having 2d fermionic particles (d is the number of discretized dimensions) for each original fermion. [2] In order to solve this problem, several strategies are in use, such as Wilson fermions, minimally doubled fermions, overlap fermions, domain wall fermions etc. Wilson fermions get rid of the doublers by adding a term (the Wilson term) to the action that explicitly breaks chiral symmetry and thus lifts the degeneracy of the doublers, giving them masses of the order of the cut-off. [3]

1.2 Introduction to FermiQCD

FermiQCD is a C++ library for fast development of parallel Quantum Field Theory (QFT) applications. Some detailed features of it are listed below:

- fully C++ (uses exceptions, streams, templates, inheritance, etc…)
- top-down design
- includes linear algebra and statistical package
- multiple lattices and fields
- automatic parallelization
- parallel random number generator
- parallel field::save and field::load methods (inherited)

- gauge_field for arbitrary SU(n)
- fermi_field for arbitrary SU(n)
- staggered_field for arbitrary SU(n) and even ndim
- Wilson, Clover, Asqtad actions (in SSE2 for SU(3)) (un-isotropic)
- Domain Wall action
- Fermilab action for heavy quarks (all dim=6 operators)
- minimum residue, stabilized bi-conjugate and uml inverters
- reads UKQCD, CANOPY, MILC and serial data formats
- easy
- safe: no need to use pointers
- flexible: can define your own fields and libraries by inheritance [4]

One of the major obstacles in solving the lattice QCD on parallel computers is that calculations of the quark interactions require very intensive computation for a highly non - local matrix determinant. Indeed, the performance of QCD code is strongly depending on the communication performance between inter - node rather than the performance of multiplication of 3 X 3 complex matrices. Since the communication pattern of the code is point – to - point communication between the nearest neighbor processors, the codes written in FermiQCD are almost ideally suited to parallel computation.

2 Materials and Methods

In this contribution we present the results of parallel calculations carried out on the BG HPC cluster, using FermiQCD, with the Wilson - Dirac operator. Quenched gauge configurations are generated with the Wilson gauge action at $\beta = 5.7, 5.85, 6$ on lattices of size 84, 124, 164. These three lattices are therefore of approximately the same physical volume. For 300 configurations at each of the three lattice sizes, Wilson quark propagators are calculated for a single point source and all color - spin combinations.

Propagators are calculated for five values of the hopping parameter κ: 0.138, 0.140, 0.142, 0.144, 0.147, corresponding to five lattice mass quarks. For the free theory the quark mass is given in terms of the lattice parameters κ and *r* as:

$$am_q = \frac{1}{2\kappa} - 4r \equiv \frac{1}{2\kappa} - \frac{1}{2\kappa_c} \tag{1}$$

with a zero at $\kappa = \kappa_c \equiv 1/8r$. For the interacting theory ($U_\mu(x) \neq 1$)we will continue to define $am_q = 1/2\kappa - 1/2\kappa_c$ with the proviso that κ_c depends on *a*. [1] To compute masses of the hadrons we define their interpolating operators (currents) [5]:

$$
\begin{aligned}
O_{\pi} &= \bar{\psi}\gamma_5\psi \\
O_{\rho,k} &= \bar{\psi}\gamma_5\gamma_k\psi \\
O_{N,\alpha} &= (\bar{\psi} \wedge C\gamma_5\psi)\psi_{\alpha} \\
O_{\Delta,k,\alpha} &= (\bar{\psi} \wedge C\gamma_k\psi)\psi_{\alpha}
\end{aligned}
\tag{2}
$$

(we suppress indices for compactness) where the wedge (vector) product is carried out in color space. The charge conjugation matrix C is defined by: $C\gamma_\mu C = -\gamma_\mu^T$ and we have chosen $C = i\gamma_2\gamma_4$. Restoring the space indices, hadron propagators are given by the correlation functions $\boldsymbol{S_{ij}:=O_iO_j^*}$, where O_i is a generic current located at site *i*. The Euclidean hadron propagators on the lattice with periodic boundary conditions, for zero momentum, can be written as:

$$
S_{t,t_0} \simeq \frac{1}{2}c_1 \cosh am_1(t - t_0 - L/2) + \frac{1}{2}c_2 \cosh am_2(t - t_0 - L/2) \tag{3}
$$

where *L* is the lattice extension in time direction, m_1 is the mass of the ground state of the particle and m_2 the mass of its first exited state. [6]

To compute the hadron propagators we have used static point sources located at t_0 = (0, 0, 0, 1) on 300 configurations. As a solver, we have used the Stabilized Biconjugate Gradient method, BiCGStab. We calculated the effective masses of π, ρ, N, Δ particles for each lattice spacing. They must reach a plateau for large time slices. Using the one - mass ansatz, the effective masses are defined by solving for am the non-linear equations:

$$
\frac{S_{t+1,1}}{S_{t,1}} = \frac{\cosh am(t - L/2)}{\cosh am(t - 1 - L/2)}, t = 1, ...L \tag{4}
$$

for mesons and

$$
\frac{S_{t+1,1}}{S_{t,1}} = \frac{\sinh am(t - L/2)}{\sinh am(t - 1 - L/2)}, t = 1, ...L \tag{5}
$$

for baryons. In order to take the continuum mass for every hadron, we perform a continuum extrapolation of the results taken for the three lattices spacing. Errors are estimated using Jackknife method. The existing codes of FermiQCD are improved and new codes need for these calculations are written.

3 Results and Discussions

Some of the results taken are presented in the figure 2, 3 and 4. In these figures are presented the calculated masses for each lattice spacing we performed simulations. These results are extrapolated in the continuum limit (a = 0) and the corresponding masses at this value of the lattice spacing, for each hadron, is the one we compare

with the experimental data. The calculated masses for the Wilson action, their errors and the experimental masses of some light hadrons (rho-meson, nucleon and delta) are given in the table 1.

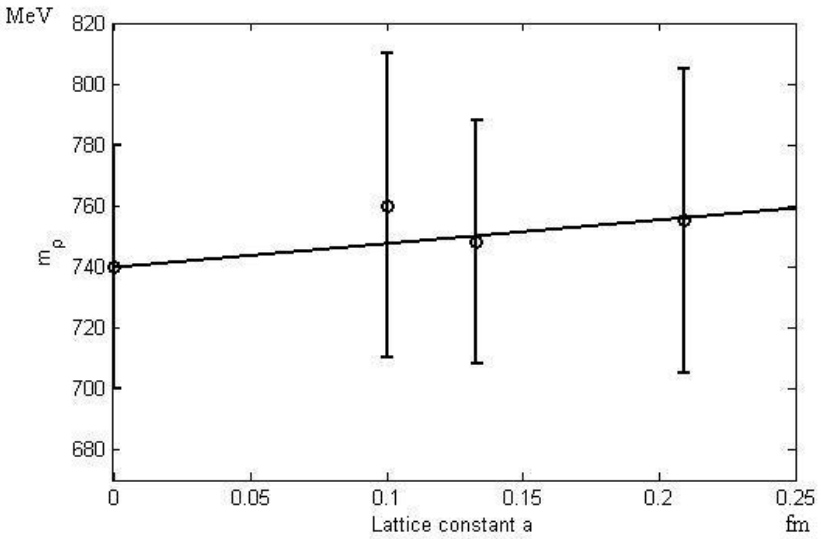


Fig. 2. The rho meson calculated mass and the corresponding errors for three lattices spacing. The continuum limit mass is taken as a result of an extrapolation.

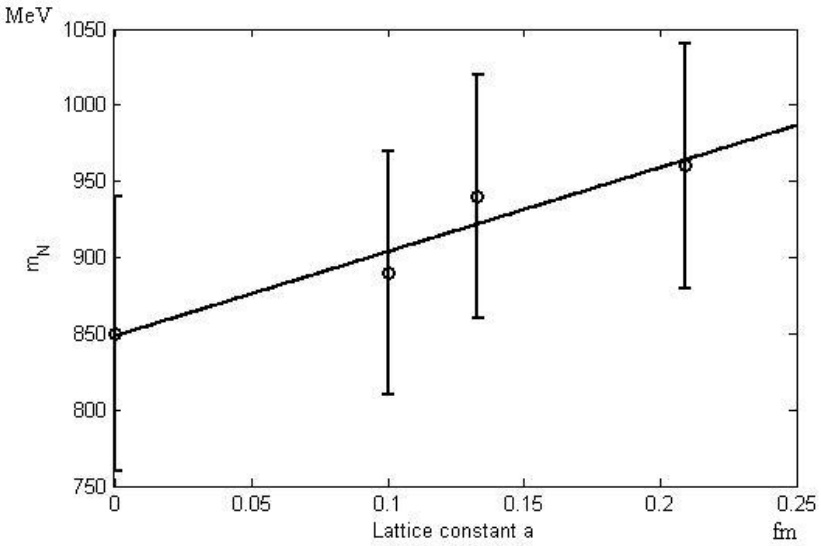


Fig. 3. The nucleon calculated mass and the corresponding errors for three lattices spacing. The continuum limit mass is taken as a result of an extrapolation.

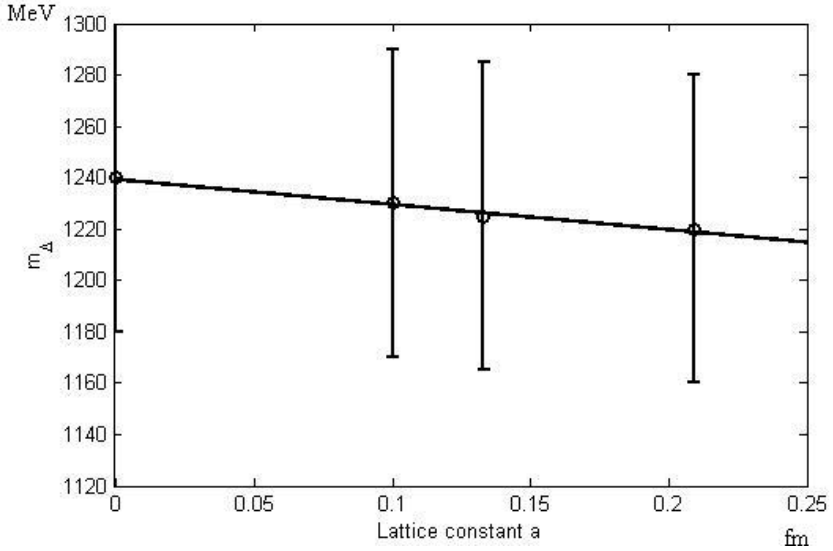


Fig. 4. The delta baryon calculated mass and the corresponding errors for three lattices spacing. The continuum limit mass is taken as a result of an extrapolation.

Table 1. Calculated masses, errors and experimental masses of light hadrons

Hadron	Mass (MeV/c^2)	Error (MeV/c^2)	Exp. Mass (MeV/c^2)
Rho - meson	750	± 60	770
Nucleon	855	± 70	938,27
Delta baryon	1240	± 87	1232

Let's see what happen with the execution time of our application when we increase the number of processors used. In Fig. 5, we illustrate the execution times of the quenched QCD application on the BG – HPC cluster for a 16^4 lattice.

To elucidate the scalability of the LQCD application we calculate the speed - up $\boldsymbol{S = T_1/T_P}$, where T_1 and T_P is the execution time for single and number of P processors, respectively. Fig. 6 shows the obtained S on the BG - HPC cluster.

As we see we have a good scalability of the application.

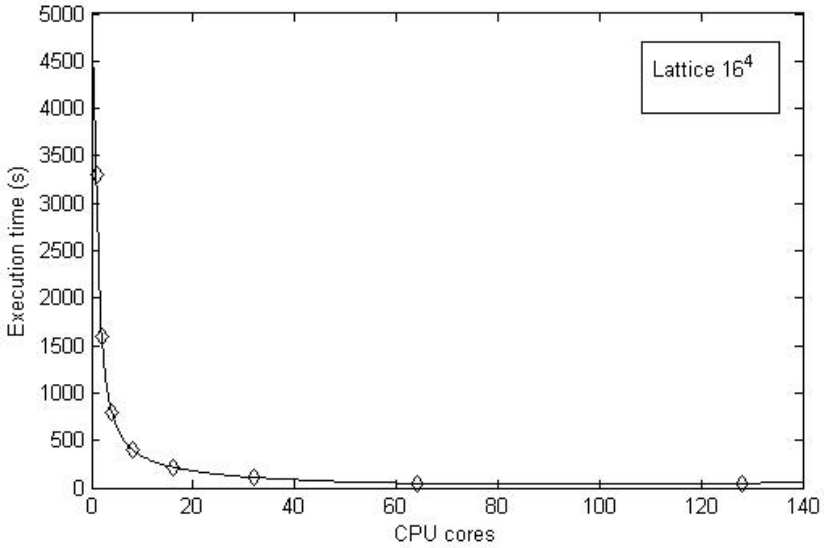


Fig. 5. Execution time of the application for different number of .cores

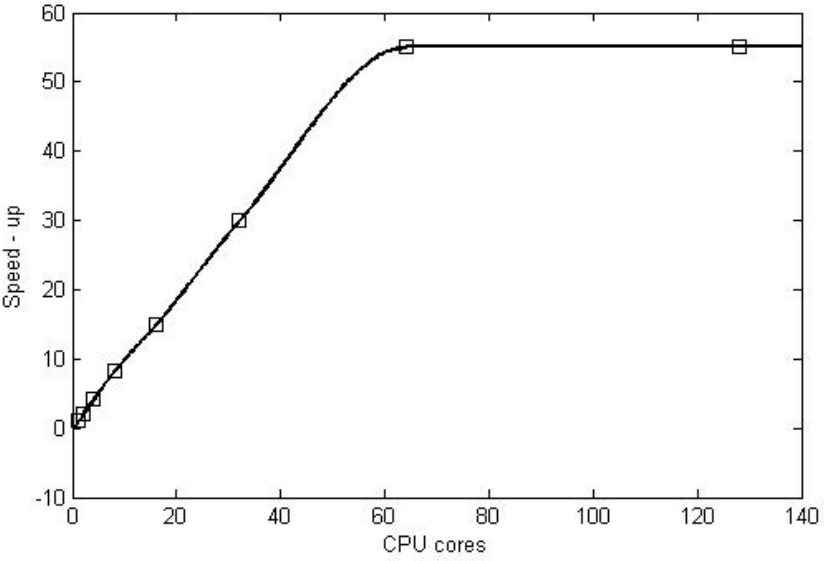


Fig. 6. The speedup of the quenched QCD for different number of cores

4 Conclusions

Hadron spectroscopy plays an important role in understanding the theory of quantum chromodynamics (QCD) and the basic lattice techniques of calculations. FermiQCD

libraries provide a tool for fast development of parallel QFT (quantum field theory) applications. It is easy, safe and flexible, in order to be used by each member of the Lattice QCD community. Parallel calculations save time and are the only way of calculations for QFT applications. Our application implemented in FermiQCD has a very good scalability. The results taken by using its gauge fields and actions (we tested Wilson action) are in agreement with the experiments at a level of 7 %, and in total agreement with the numerical simulations done until now with other softwares and techniques. Statistical errors when autocorrelations are present, can't be estimated using standard methods, because they give a lower error than the real one. Non – parametric methods, like Jackknife method give a real estimation of statistical errors.

Acknowledgments. This work makes use of results produced by the High - Performance Computing Infrastructure for South East Europe's Research Communities (HP-SEE), a project co-funded by the European Commission (under contract number 261499) through the Seventh Framework Program. HP-SEE involves and addresses specific needs of a number of new multi-disciplinary international scientific communities (computational physics, computational chemistry, life sciences, etc.) and thus stimulates the use and expansion of the emerging new regional HPC infrastructure and its services. Full information is available at http://www.hp-see.eu/.

References

1. Gupta: Introduction to lattice QCD (1998) arXiv:hep-lat/9807028
2. Nieslen, Ninomiya: Absence of Neutrinos on a Lattice. Nucl. Phys. B 185, 20 (1981) Bibcode 1981NuPhB.185...20N, doi:10.1016/0550-3213(81)90361-8; Nieslen, Ninomiya: No go theorem for regularizing chiral fermions. Phys. Lett. B 105, 219 (1981) Bibcode 1981PhLB..105..219N, doi:10.1016/0370-2693(81)91026-1
3. Creutz, M.: Quarks, gluons and lattices. Cambridge University Press (1985)
4. `http://www.fermiqcd.net`
5. Fucito, F., et al.: Nucl. Phys. B 210, 407 (1982)
6. Boriçi, A.: Krylov Subspace Methods in Lattice QCD. PhD thesis (1996)

Self-Avoiding Hamiltonian Walks Counting in Parallel Processing Mode

Igor Ševo[1], Sreten Lekić[2], and Mihajlo Savić[1]

[1] Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina
[2] Faculty of Natural Sciences and Mathematics, University of Banja Luka, Bosnia and Herzegovina
badaboom@etfbl.net

Abstract. We have developed a program for counting self-avoiding Hamiltonian walks to run on multiple processors in a parallel mode. We study Hamiltonian walks (HWs) on the family of two-dimensional modified Sierpinski gasket fractals, as a simple model for compact polymers in nonhomogeneous media in two dimensions. We apply an exact recursive method which allows for explicit enumeration of extremely long Hamiltonian walks of different types: closed and open, with end-points anywhere in the lattice, or with one or both ends fixed at the corner sites. The leading term n is characterized by the value of the connectivity constant 1, which depends on fractal type, but not on the type of HW.

Keywords: hamiltonian walks, fractal, parallel processing.

1 Introduction

Self-avoiding walks (SAWs) have been used in the modeling of polymers, micro magnetics configuration, studies of diffusion through the porous media, statistics of polymer chains configuration in solution [1]. The most intensive preoccupations are directed toward the polymer phase transitions. Due to excluded volume effect, at high temperatures T (good solvent) long polymer chains are in swollen configurations. At low temperatures (poor solvent) polymers are in a collapsed state, caused by the attractive interactions of different sections of a polymer, mediated by a solvent. The transition between these two states occurs at the θ temperature, at which excluded volume and attractive forces balance. Whereas the swollen and θ phases has been well investigated by now, the entropy scaling of the collapsed phase is still an open issue. A closely related problem is the scaling of Hamiltonian walks (HWs), which are SAWs that visit all the sites of the underlying lattice [2]. HWs represent the $T = 0$ limit of collapsed polymers, and they are also used in the studies of polymer melting, as well as in the context of protein folding [3]. The number Z_N of HWs on homogeneous lattices for large N behaves as $Z_N \sim \omega^N \mu^{N_S^\sigma} N^\alpha$ where $\sigma = (d-1)/d$, (d is the dimensionality of the lattice), μ is constant less than 1, and ω is the connectivity

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_7, © Springer International Publishing Switzerland 2014

constant. Proposed scaling form for HWs differs from the ordinary SAW case (swollen polymer), where average number of N-step SAWs, for large N, behaves as $\omega^N N^\alpha$, and critical exponent α depends only on d (which is not the case for HWs). The term $\mu^{N_S^\sigma}$ is result of the exact study of HWs on the Manhattan lattice [4], and the consequence of conjecture that collapsed polymer (globule) has a sharp boundary and surface tension terms are more dominant [5].

2 Hamiltonian Walks on Sierpinski Gasket Fractals

SG is a well known fractal lattice, which can be constructed recursively, starting with the generator (gasket of order $r = 1$), which consists of three unit equilateral triangles, arranged to form a larger triangle (see Fig. 1). The subsequent fractal stages are constructed self-similarly, by replacing each of the unit triangles of the initial generator with a new generator. To obtain the rth-stage fractal lattice (rth order gasket), this process of construction has to be repeated $(r-1)$ times, and the complete fractal is obtained in the limit $r \to \infty$ numbers of sites on the rth order gasket is equal to $N_r = \frac{3}{2}(3^r + 1)$. SG resembles 3-simplex lattice and indeed has the same fractal dimension $df = \ln 3/\ln 2$. An open HW on a third order gasket is shown on Fig. 3, together with its coarse-grained versions. Comparing with Fig. 1 one can observe that larger number of types of possible HW configurations exists on SG than in the case of 3-simplex lattice. There are exactly eight different types of walks, and they are depicted on Given-Mandelbrot fractals represent the fractal family characterized with integer $b \geq 2$ (scaling factor) [7]. The first element of row is Sierpinski gasket, fractal with $b = 2$. At the same way, other fractal are constructed k. Initiator is a site triangle a. Aa the first step of construction, $r = 1$, $\frac{b(b+1)}{2}$ initiator sets in equilateral triangle ba on this way that the vertices are connected.

Constructed structure represents the generator of order one $G^{(1)}(b)$. In the second steep, $r = 2$, $\frac{b(b+1)}{2}$ initiator set on the same way in equilateral triangle b^2a site. This is the generator of order 2. After r iterative steps generator order r, $G^{(r)}(b)$, triangle with $b^r a$ site has $\frac{b(b+1)}{2}$ generators $(r-1)$ order, and complete fractal when $r \to \infty$. The first 3 generators of GM fractal $b = 3$ parameter is on fig 1.

Fractal dimension is

$$d_f = \frac{\ln \frac{b(b+1)}{2}}{\ln b}\,. \tag{1}$$

Initiator vertexes and sites of GM fractal make fractal lattice. The number of nodes, N_r, of $G^{(r)}(b)$, is recursive expressed as $N_r = \frac{b(b+1)}{2}(N_{r-1} - 3) + \sum_1^{b+1} k$. The first term expresses number of nodes of all $\frac{b(b+1)}{2}$ generators of order $r-1$, with vertices exception. Summa presents the number of generators order $r-1$, and is equal $\sum = (b+1)(b+2)/2$. After simplification N_r expression for

$N_r = \frac{b(b+1)}{2} N_{r-1} - b^2 + 1$, and iteration toward the begin, $N_0 = 3$, and

$$N_r = \frac{b+4}{b+2}\Big(\frac{b(b+1)}{2}\Big)^r + 2\frac{b+1}{b+2}\,. \tag{2}$$

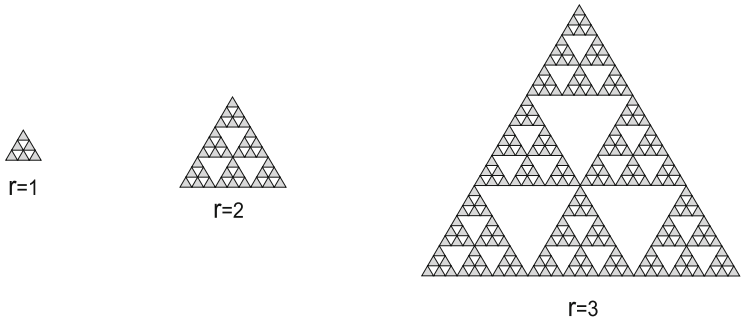


Fig. 1. Construction GM fractal by $b = 3$

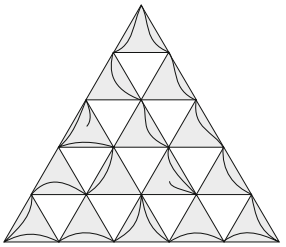


Fig. 2. Open HW on generator $(r + 1)$- order for GM fractal $b = 5$. HW consists od one walk A, one A_1, four B and nine walks B_1 on coarsed generators of order r, and this is the shape $AA_1B^4B_1^9$ in reduced notation.

The corresponding numbers of HWs on the rth-2 order gasket will be denoted by $A^{(r)}$, $A_1^{(r)}$,$A_1^{(r)}$ $B^{(r)}$,$B_1^{(r)}$,$C^{(r)}$, $D^{(r)}$ and $D_1^{(r)}$.

These numbers fulfill the following recursion relations: Since in [6] it was exactly shown that $B(r) = const \cdot \omega^{N_r}$, with $\omega = 12^{1/9}$, one finally obtains the same scaling form as for 3-simplex lattice: $Z_0(r) = \omega^{N_r} N_r^{\gamma}$, with the same value of exponent $\gamma = \ln 16/\ln 3$. It was also shown in [8] that the overall number Z(r) C of closed HWs on SG lattice scales according to the formula ω^{N_r}, again the same as in the case of 3-simplex. The equality of exponents for these two fractal lattices is in accord with the fact that SAWs on them belong to the same universality class [10]. On the other hand, it is known that exponents for HWs on different 2d Euclidean lattices have different values [11], which is explained to be a consequence of the frustration, induced by the strong constraint that all the sites must be visited.

It is believed that a relevant physical measure of this frustration is the number of contacts per monomer, i.e. vertex pairs which are not adjacent along the HW, but are the nearest neighbors on the lattice. Nevertheless, the number of contacts on 3-simplex is one, whereas it is two on SG, so that one could have expected different values of γ. In order to gain a deeper insight into the problem

of universality and frustration of HWs on lattices embedded in 2d space,the asymptotic behavior of HWs on the appropriate generalizations of 3-simplex and SG fractals was analyzed [12].

3 Hamiltonian Walks on Given-Mandelbrot Fractals

One way to construct the SG fractal is to start with a generator that consists of $b(b+1)/2$ unit equilateral triangles, arranged to form a b times larger triangle. Enlarging the generator b times and substituting the smallest triangles with the generator, and then repeating this procedure recursively, one obtains fractal lattice characterized with the integer b. For b = 2, 3, ... the complete so called Given-Mandelbrot (GM) family of fractals is obtained [9]. SG is the first member of this family, with the scaling parameter $b = 2$.

Number of open HWs on the $(r+1)$th stage of any GM fractal construction can be expressed in terms of numbers of 8 HW types within the rt-h order stage, in a similar manner as in the case of SG. Recursion relations for numbers of B, and B1-type walks on two successive stages of fractal construction have the following form

$$B' = pB^{\frac{b(b-1)}{2}+1}B_1^{b-1}, \qquad B_1' = B_1^b \tag{3}$$

as was shown in [6]. whereas the numbers of one-leg configurations: A, A1, A2, and C, satisfy a closed set of recursion relations, which can be put in matrix form (9) where a_{ij} are polynomials in $B(r)$ and $B(r)_1$.

The number $Z(r+1)$ of open HWs on the gasket of order $(r+1)$, can be expressed as (4) where we have suppressed index r on the right-hand side of this relation, and K_{ij} are numbers that depend only on b. Substituting established asymptotical behavior of A_i, C and D_i in the latter expression, one finds that all terms on the right-hand side of equation have the same asymptotical form. The values of γ for are equal to $2 \le b \le 8$ are 2.5237. . ., 2.1841 . . ., 2.3411 . . ., 2.2461 . . ., 2.2981 . . ., and 2.2755 . . ., respectively. One should mention here that number of closed HWs asymptotically behaves as Z(r) as was established in [6], where also a closed formula was derived [8].

$$\begin{aligned}\mathcal{Z}_{r+1} = &\sum_{i\le j=0}^{2} k_{ij}A_{i\,r}A_{j\,r}B_r^{m-2+(i+j)}B_{1\,r}^{n-(i+j)} + C_r\sum_{i=0}^{2} k_i A_{i\,r}B_r^{m-1+i}B_{1\,r}^{n-1-i} \\ &+ k_3 C_r^2 B_r^m B_{1\,r}^{n-2} + \sum_{i=0}^{1} k_i{}' D_{i\,r}B_r^{m+i}B_{1\,r}^{n-1-i}\,,\end{aligned} \tag{4}$$

where $m = b$ and $n = \frac{b(b-1)}{2}$. Term coefficients k_{ij}, k_i ..., are independent of r, doesn't influence on HW's number.

Recurrent relations B i B_1 are

$$B_{r+1} = p\,B_r^m B_{1\,r}^n\,, \quad B_{1\,r+1} = p\,B_r^{m-1}B_{1\,r}^{n+1}\,, \tag{5}$$

and for $A\,, A_1\,, A_2$ and C :

$$\begin{aligned} A_{i\,r+1} &= \sum_{j=0}^{2} a_{ij} A_{j\,r} B_r^{m-1-(i-j)} B_{1\,r}^{n+(i-j)} + a_{i3} C_r B_r^{m-i} B_{1\,r}^{n-1+i}\,, \quad i = 0, 1, 2\,, \\ C_{r+1} &= \sum_{i=0}^{2} c_i A_{i\,r} B_r^{m-2+i} B_{1\,r}^{n+1-i} + c_3 C_r B_r^{m-1} B_{1\,r}^{n}\,. \end{aligned} \tag{6}$$

Term coefficients in recurrent relations for A_1 and A_2 can expressed over the coefficients A and B, $a_{10} = a_{00} - p/2\,, a_{11} = a_{01}\,, a_{12} = a_{02} + p/2\,, a_{13} = a_{03}$ i $a_{20} = a_{00} - p\,, a_{21} = a_{01}\,, a_{22} = a_{02} + p\,, a_{23} = a_{03}$, .

The number od B and B_1 are proportional to r. division of this relation give $\frac{B_{1\,r+1}}{B_{r+1}} = \frac{B_{1\,r}}{B_r}$, and $\frac{B_{1\,r}}{B_r} = const = t = \frac{B_{1(1)}}{B_{(1)}}$ (t is the rate of starting number of walks B_1 i B, i and is the function of fractal parameter b). One variable $A_{i\,r}$ can be expressed over other $A_{2\,r} = 2tA_{1\,r} - A_r t^2$, $\forall\ r > 1$. Elimination $B_{1\,r}$ i $A_{2\,r}$ relations in (5) and (6) go in

$$\begin{aligned} B_{r+1} &= p t^n B_r^{m+n}\,, \\ A_{r+1} &= B_r^{m+n+1}\,[t^n(a_{00} - a_{02})A_r + t^{n-1}(a_{01} + 2a_{02})A_{1\,r} + t^{n-1} a_{03} C_r]\,, \\ A_{1\,r+1} &= B_r^{m+n+1}\,[t^{n+1}(a_{00} - a_{02} - p)A_r + t^n(a_{01} + 2a_{02} + p)A_{1\,r} + t^n a_{13} C_r]\,, \\ C_{r+1} &= B_r^{m+n+1}\,[t^{n+1}(c_0 - c_2)A_r + t^n(c_1 + 2c_2)A_{1\,r} + t^n c_3 C_r]\,. \end{aligned} \tag{7}$$

Iteration of the first of these relations, or go with solution $B_r = c\, c_1^{(m+n)^r}$), and(2), and definitions t, m i n, the result is[1]

$$B_r = \mathfrak{B}\omega^{N_r}\,, \quad \text{gdje je} \quad \omega = \left[p\right]^{k_1} \left[B_{(1)}\right]^{k_2} \left[B_{1(1)}\right]^{k_3}\,, \quad \text{i} \tag{8}$$

$$\begin{pmatrix} x_{r+1} \\ x_{1\,r+1} \\ y_{r+1} \end{pmatrix} = \frac{1}{p} \begin{pmatrix} a_{00} - a_{02} & (a_{01} + 2a_{02})/t & a_{03}/t \\ t(a_{00} - a_{02} - p) & a_{01} + 2a_{02} + p & a_{03} \\ t(c_0 - c_2) & c_1 + 2c_2 & c_3 \end{pmatrix} \begin{pmatrix} x_r \\ x_{1\,r} \\ y_r \end{pmatrix}\,, \tag{9}$$

4 Implementation and Parallelization

Initial attempts to find the number of walks was based on sequential Fortran code which, while providing correct results, left a lot to be desired both in performance and scalability department. In order to obtain desirable performance and ease the parallelization we opted to forgo brute-force path counting approach and try to find a more elegant solution.

The problem of building all walks on the gasket can be reduced to the problem of constructing each walk using the predefined elementary triangles. Since, given

[1] $\mathfrak{B} = p^{-\frac{2(b+2)}{b(b-1)(b+4)}} [B_{(1)}]^{\frac{b(b+2)-4}{b(b+4)}} [B_{1(1)}]^{-\frac{b+2}{b+4}}$.

Table 1. Coefficients p, a_{0i} and c_i in recursion relations for HW B, A i C for the numbers of B, A i C -type, found by direct computer enumeration of all possible corresponding configurations, together with the values of connectivity constant and exponent on GM fractals with $2 \leq b \leq 8$

b	a_{01}	a_{02}	a_{03}	c_0	c_1	c_2	c_3
2	2	0	2	1	2	1	6
3	14	4	22	12	16	4	32
4	128	82	212	152	168	48	352
5	1532	1482	2704	2544	2120	424	4048
6	23812	31518	42368	52072	35152	6568	67680
7	486284	817798	878168	1340536	735312	112088	1374944
8	12778136	26422308	23141696	43647128	20117360	2742936	37493824
9	438476480	1079408072	797598000	1797330104	705340848	167728048	1301033984

a division, a gasket can consists of a constant number of sub triangles, the gasket itself can be constructed by combining the available triangles.

The algorithm used here maps the elementary walks to elementary triangles and combines them in a lower triangular matrix of elementary triangles, thereby mimicking the above given structure. As the elementary walks are predefined, the elementary triangles are defined accordingly. In this way, there exist six predefined triangles, each constructed from a separate elementary walk, to be used for constructing the gasket. The division of the gasket determines the dimensions of the lower triangular matrix, which is a square matrix, hence both dimensions are equal, and the total number of triangles to use in constructing the gasket.

The algorithm attempts to build a gasket by placing elementary triangles in the triangular matrix. The process starts from the top-most, left-most cell of the triangular matrix and proceeds recursively to build the gasket. Each call of the recursive function places a single triangle at the position of the call and proceeds to call the same function on the next position in the triangular matrix. The recursive calling ends when either the last cell of the matrix is reached or no suitable elementary triangle can be placed at the specified position.

Elementary triangles are arranged in such a way that the HWs extruding from previously placed elementary triangles are continued by newly placed elementary triangle rotated and/or mirrored to fit in an adequate way to satisfy the continuation requirement.

After a gasket is constructed, a final check is performed to determine whether the gasket is of the targeted type. This check is performed in such a way that three points of the triangle are checked for matches against the target triangle. If they match, the gasket is checked for loops since certain triangle combinations can result in loops. If no loops exist, the given gasket is determined to be of the targeted type and the corresponding HW counter is increased.

Optimization of the algorithm is reflected in the triangle-building approach, as certain paths are eliminated, which would otherwise be unnecessarily evaluated. The approach does introduce the path looping problem which can efficiently be dealt with by a final check of looping. This check is done in linear time and is executed by a single thread.

4.1 Parallelization and Scalability

We have implemented two approaches to parallelization of aforementioned implementation. Both approaches are based on OpenMP multi-threaded shared memory model[13].

The first approach is relatively simple but comes with some caveats as it is based on OpenMP nested parallelism. Upon calling a function on a certain cell of the matrix, a set of maximum of six elementary triangles can be placed at the given position. For each of these possible placements a new thread is spawned and the current state of the triangular matrix copied and assigned to it. This way, the number of threads increases exponentially with each call of the recursive function. Certain constructs of triangles can efficiently be resolved by a single thread, so for these constructs only a single thread is used (this refers to the non-necessity of branching for constructing the final n triangles). There are two issues with this approach. First one is the fact that just creating a large number of threads negatively impacts performance, event though not all threads are active at the same time. Second issue is that if we try to keep number of simultaneously active threads, as well as newly created threads, close to a reasonable number we pay the penalty close to the end of calculation when few long-lasting threads effectively reduce the scalability of the implementation. Careful fine-tuning allowed us to achieve very good scalability for up to 48 CPU cores but at the price of creating several thousands of threads.

While nested parallelism approach did produce scalability, it also caused severe problems with different compilers, versions and architectures available at various HP-SEE HPC resource centers. HP-SEE project represents a continuation of series of GRID and HPC related projects in SEE region [14]. In order to alleviate these problems we implemented a version that uses OpenMP 3.0 task construct. This allowed us to keep the number of threads close to the number of physical CPU cores while improving the scalability (Table 2).

Scalability testing was performed at Pecs SC resource center. Pecs supercomputer is a SGI 1000 Ultraviolet supercomputer based around Intel Xeon X7542 6-core processors with ccNUMA SMP architecture. The application was compiled by Intel C++ compiler version 12.1.5 and GCC C/C++ compiler version 4.3.4 both with -O3 optimization level. While GCC did produce measurably

Table 2. Scalability testing of SFHG application

CPU cores	Walk type	Level	CPU time/Wall time	Efficiency
1	C	8	1.00	1.00
2	C	8	1.97	0.99
4	C	8	3.91	0.98
8	C	8	7.82	0.98
12	C	8	11.31	0.94
16	C	8	15.02	0.94
24	C	8	21.74	0.91
48	C	8	40.93	0.85

better results for nested parallelism version, incomplete support for OpenMP task construct in available version prevented us from comparing the compilers for our production code.

5 Summary and Conclusion

We have analyzed asymptotic behavior of the numbers of open and closed Hamiltonian walks on Given-Mandelbrot (generalized Sierpinski gasket) fractal families and made a few new steps (8 and 9) in exact evaluating the numbers of self-avoiding walks and calculating the asymptotic behavior of leading terms. Potential capability of program in parallel processing will give the analyzing tool for more complex 3D fractal structures.

Acknowledgement. This work makes use of results produced by the High-Performance Computing Infrastructure for South East Europe's Research Communities (HP-SEE), a project co-funded by the European Commission (under contract number 261499) through the Seventh Framework Programme. HP-SEE involves and addresses specific needs of a number of new multi-disciplinary international scientific communities (computational physics, computational chemistry, life sciences, etc.) and thus stimulates the use and expansion of the emerging new regional HPC infrastructure and its services.

References

1. Vanderzande, C.: Lattice Models of Polymers. Cambridge University Press (1998)
2. Madras, N., Slade, G.: The Self Avoiding Walk. Birkäuser (1993)
3. Kondev, J., Jacobsen, J.L.: Phys. Rev. Lett. 81, 2922 (1998)
4. Duplantier, B., David, F.: J. Stat. Phys. 51, 327 (1988)
5. Owczarek, A.L., Prellberg, T., Brak, R.: Phys. Rev. Lett. 70, 951 (1993)
6. Stajić, J., Elezović-Hadžić, S.: J. Phys. A: Math. Gen. 38, 5677 (2005)
7. de Gennes, P.G.: Scaling Concepts in Polymer Physics. Cornell University Press (1979)
8. Elezović-Hadžić, S., Marčetić, D., Maletić, S.: Phys. Rev. E 76, 011107 (2007)
9. Given, J.A., Mandelbrot, B.B.: J. Phys. A: Math. Gen. 16, L565 (1983)
10. Dhar, D., Singh, Y.: Statistics of Polymers in Random Media. World Scientific, Singapore (2005)
11. Bradley, R.M.: J. Phys. A: Math. Gen. 22, L19 (1989)
12. Orland, H., Itzykson, C., de Dominicis, C.: J. Physique 46, L353 (1985)
13. OpenMP, `http://openmp.org/wp/`
14. Balaz, A., Prnjat, O., Vudragovic, D., Slavnic, V., Liabotis, I., Atanassov, E., Jakimovski, B., Savic, M.: J. Grid Comput. 9, 135 (2011)

Conformational Analysis and HF *ab initio* Geometry Optimization of Kyotorphine and Its Sulfo-Analogues Norsulfoarginine-Tyrosine and Tyrosine-Norsulfoarginine

Nicolay I. Dodoff, Tatyana A. Dzimbova, and Tamara I. Pajpanova

Acad. RoumenTsanev Institute of Molecular Biology,
Bulgarian Academy of Sciences,
Acad. G. Bonchev Street, Block 21, 1113 Sophia, Bulgaria
{dodoff,taniadz,tamara}@bio21.bas.bg

Abstract. Molecular mechanics (MM) conformational search has been performed for the molecules of the endogenous dipeptide kyotorphin (kyo) and its synthetic sulfo-analogues norsulfoarginine-tyrosine (NsArg-Tyr) and tyrosine-norsulfoarginine (Tyr-NsArg). The MM-found minimum-energy conformations were further optimized at HF *ab initio* level (3-21G* basis set) in gas phase and in water medium. The non-ionic and the *zwitter*-ionic forms and the *E*- and *Z*-diastereomers of the dipeptides were considered. In all cases, all the lowest-energy conformations adopt a specific *scorpion*-like conformation with close proximity between the guanidino and phenolic residues. The relative enegies of the different forms, the geometric parameters and the role of the intramolecular hydrogen bonding in stabilizing the structures are discussed.

Keywords: kyotorphin, neuropeptide, sulfo-analogues, conformational analysis, HF *ab inito* geometry optimization.

1 Introduction

The dipeptide kyotorphin (Tyr-Arg, Kyo) plays a role in pain modulation in the mammalian central nervous system (CNS), and is one of the most investigated neuropeptides. The Tyr-Arg motif exists widely throughout the brain not only as kyotorphin, but also as the N-terminal part of several endogenous analgesic peptides [1, 2]. Also, this peptide is very rapidly degraded by aminopeptidases [3]. One of the successful strategies in the design of neuropeptides with enhanced stability and improved delivery to the CNS is that with the use of non-protein amino acids, like canavanine (Cav), a structural analogue and antimetabolite of arginine (Arg). In our previous in vivo experiments we demonstrated that Tyr-Cav exerted a strong-reversible analgesic effect, more pronounced than that of Kyo. Bearing in mind these and the fact that norsulfoarginine (NsArg) [4] is a structural analogue of arginine and canavanine, we synthesized a series of new peptides with expected analgesic activity, containing NsArg residues in their molecules: NsArg-Tyr, Tyr-NsArg, Tyr-NsArg-NH_2 and

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/ 978-3-319-01520-0_8, © Springer International Publishing Switzerland 2014

Tyr-NsArg-Obzl [5 ,6]. The conformational features of these dipeptides are of particular interest, both from theoretical and pharmacological points of view. Since no single-crystal X-ray diffraction data for the compounds are available until now, we undertook a quantum-chemical modelling of their structure. Here we present our preliminary computational results for Kyo, NsArg-Tyr and Tyr-NsArg.

2 Computational

The initial conformational search for each molecule (finding the minimum-energy conformations (MECs), and among them – the lowest-energy conformation (LEC)) was performed at Molecular Mechanics level (MM+, force field; bond charges) with varying the torsion angles around the rotatable bonds (HyperChem 7.5 programme [7], Directed search, RMS gradient < 0.009). The lowest-energy conformation (LEC) for each diastereomer was further minimized with the same force field (RMS < 0.001). The LECs thus obtained were then subjected to Hartree-Fock (HF) *ab initio* optimization in gas phase and in water medium (PCM) with 3-21G* basis set [8]. Maximum gradient < $1 \cdot 10^{-4}$ Hartree/Bobr (< $9 \cdot 10^{-4}$ for NsArg-Tyr-*E*-W and NsArTyr-*Z*-W). The HF calculations were performed on a personal computer, as well as on a HPC Cluster Platform Express 7000 (Institute of Parallel Processing, BAS) using the Firefly [9], and GAMESS [10] quantum chemistry packages, respectively.

3 Results and Discussion

For kyotorphine (kyo), the non-ionic *E*- and *Z*-diastereomers with respect to the configuration of the C1=N1 and C2-C3 bonds, as well as the zwitter ionic form (guanidinium-carboxylate) were considered. The HF ab initio optimized LECs with the atom labeling scheme are depicted in Fig. 1, and in Table 1 their energies and geometric parameters are collected. The most stable is the *zwitter* ionic form in water medium (kyo-zw-W). Among the non-ionic forms, the *Z*-diastereomer is more stable. In all cases the molecule is more stable in water medium than in free state (gas phase). For the kyotorphin analogues norsulfoarginine-tyrosine (NsArg-Tyr) and tyrosine-nosulfoarginine (Tyr-NsArg), again the non-ionic *E*- and *Z*-diastereomers, as well as the zwitter ionic (amonium-carboxylate) *E*- and *Z*- forms were regarded (Figs. 2 and 3). The *zwitter* ionic forms, upon HF optimization in gas phase, were found to converge to the corresponding non-ionic diastereomer (NsArg-Tyr-*E*, NsArg-Tyr-*Z*, Tyr-NsArg-*E* and TyrNsArg-*Z*). This means that at the level of theory applied, the *zwitter* ionic forms do not correspond to minima on the potential energy surface. For water medium, however, equilibrium conformations were indeed obtained (NsArg-Tyr-*E*-*zw*-W, NsArg-Tyr-*Z*-*zw*-W, Tyr-NsArg-*E*-*zw*-W and Tyr-NsArg-*Z*-*zw*-W). In distinct with kyo, the most stable forms of the sulfo analogues appeared the non-ionic Z-diastereomers in water (NsArg-Tyr-*Z*-W and Tyr-NsArg-*Z*-W), whereas the *zwitter* ionic forms (in water) exhibited higher (*ca.* 0.05 and 0.03 Hartree, respectively) energies. Tyr-NsArg isomer is more stable than NsArg-Tyr, *e. g.* the energy difference

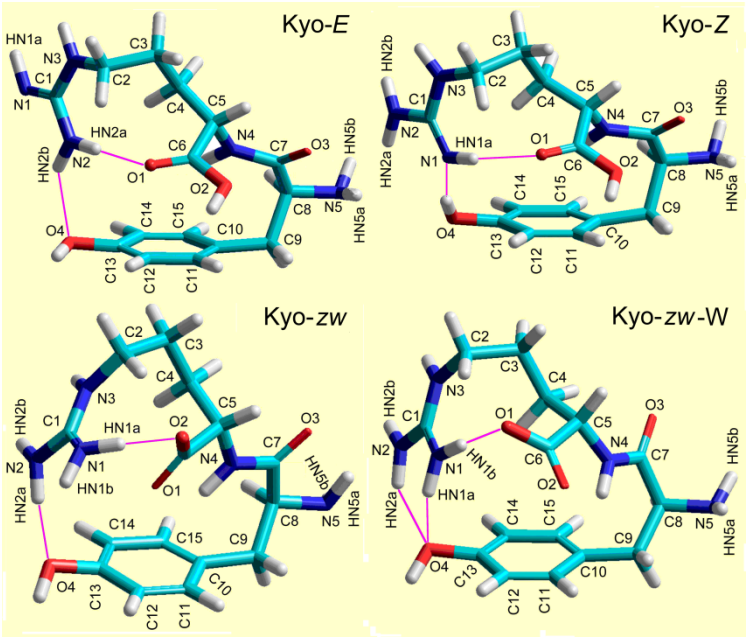


Fig. 1. HF *ab initio* optimized LECs for the neutral (kyo-*E* and kyo-*Z)* kyotorphine*E*- and *Z*-diastereomersand the *zwitter*-ionic form of in gas phase (kyo-*zw*) and water solution (kyo-*zw*-W) with atom labeling and H-bonds

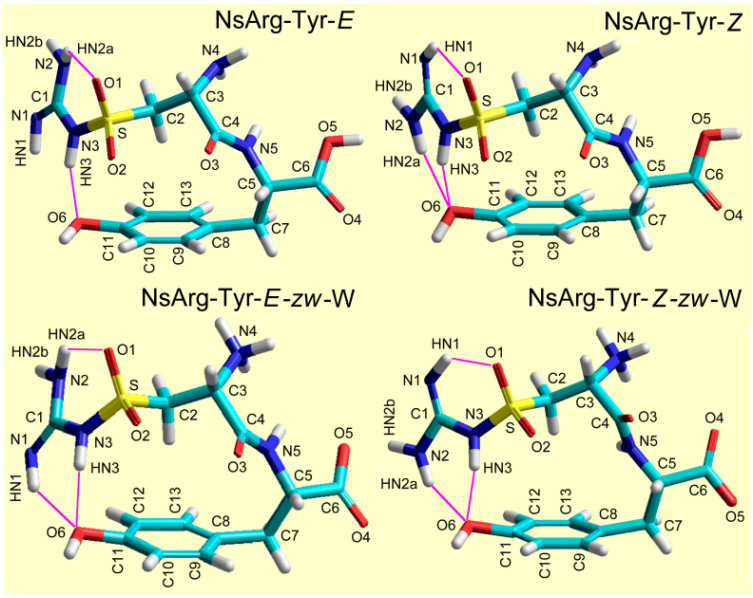


Fig. 2. HF *ab initio* optimized LECs for the neutral (NsArg-Tyr-*E*, NsArg-Tyr-*Z*) and *zwitter*-ionic (NsArg-Tyr-*E*-*zw*-W andNsArg-Tyr-*Z*-*zw*-W) forms of norsulfoarginine-tyrosine*E*- and *Z*-diastereomers in gas phase and water solution (W) with atom labeling and H-bonds

Table 1. Eenergy differences (*Δ*E) and selected geometric parameters for kyotorphin diastereomers in neutral and *zeitter*-ionic form in gas phase and in water medium (see Fig.1)

Geometric parameter	Molecule					
	Kyo-*E*[a]	Kyo-*E*-W	Kyo-*Z*	Kyo-*Z*-W	Kyo-*zw*	Kyo-*zw*-W
			*Δ*E, Hartree[b]			
	0.0581	0.0194	0.0414	0.0077	0.0577	0.0000
			Bond lengths, Å			
N1-C1	1.274	1.285	1.277	1.289	1.333	1.325
N2-C1	1.369	1.367	1.375	1.364	1.346	1.336
C1-N3	1.389	1.377	1.373	1.365	1.328	1.327
N3-C2	1.464	1.469	1.469	1.470	1.489	1.484
C2-C3	1.552	1.550	1.549	1.547	1.558	1.544
O4-C13	1.384	1.389	1.370	1.379	1.404	1.405
C5-C6	1.516	1.516	1.519	1.520	1.528	1.552
C6-O1	1.210	1.211	1.209	1.210	1.253	1.245
C6-O2	1.350	1.344	1.350	1.347	1.265	1.264
			Sum of bond angles at C and N atoms of guanidine fragment, deg			
Σang (C1)	360.0	360.0	360.0	360.0	359.8	360.0
Σang(N1)					345.9	358.6
Σang(N2)	359.8	357.8	355.3	359.5	351.4	359.4
Σang(N3)	359.7	359.8	359.6	359.9	358.9	359.9
			Torsion angles, deg			
C1-N3-C2-C3	104.3	105.0	99.3	102.0	128.5	97.5
C11-C10-C9-C8	-108.0	-111.1	-103.8	-105.9	-121.8	-137.2
O4-C13-C12-C11	-176.5	-176.4	-175.5	-176.8	-172.5	-173.3
C12-C11-C10-C9	174.2	175.6	170.0	171.4	174.2	175.5
			H-bonding geometry, Å and deg			
N1-HN1a	0.998	1.000	0.995	0.996	1.022	0.992
HN1a···O1			2.196	2.157		
N1-HN1a···O1			158.0	168.0		
HN1a···O2					1.723	
N1-HN1a···O2					151.4	
N1-HN1b					0.990	1.011
HN1b···O1						1.764
N1-HN1b···O1						171.8
HN1a···O4						2.186
N1-HN1a···O4						139.3
O4-HO4	0.940	0.943	0.969	0.977	0.941	0.944
HO4···N1			1.776	1.736		
O4-HO4···N1			160.9	168.2		
N2-HN2b	0.989	0.991	0.987	0.987	0.989	0.989
HN2b···O4	2.259	2.188				2.059
N2-HN2b···O4	143.4	149.9				145.4
N2-HN2a	0.987	0.990	0.989	0.987	1.000	0.995
HN2a···O1	2.045	1.984				
N2-HN2a···O1	173.5	177.4				
HN2a···O4					2.076	2.059
N2-HN2a···O4					161.5	145.4

[a] *E* and *Z* denote the configuration of the N1=C1 bond with respect to N3–C2 bond.[b] The absolute energies are: -1146.8892, -1146.9279, -1146.9059, -1146.9396, -1146.8896 and -1146.9473 Hartree, respectively.

between the Z-diastereomers in water is 0.013 Hartree (34 kJ/mol) in favour of the later (Tables 2 and 3).

It should be emphasized that in the three cases (kyo and its two isomeric sulfoanalogues), all the LECsadopt a *scorpion*-like conformation with close proximity of the guanidino group and phenolic residue. This finding is important because it clearly demonstrates that the molecules of the artificial isomeric dipeptides NsArg-Tyr and Tyr-NsArg are indeed sterically similar to that of kyotorphin. All the LECs are qualitatively equivalent and have similar values of the geometric parameters (bond lengths, bond angles and torsion angles), as seen from Tables 2 and 3. There is a regularity that in the *E*-diastereomers of the sulfo analogues, the NH_2 group of the guanidine residue has a flattened pyramidal configuration, whereas in the Z-distereomers, it is fully planarized.

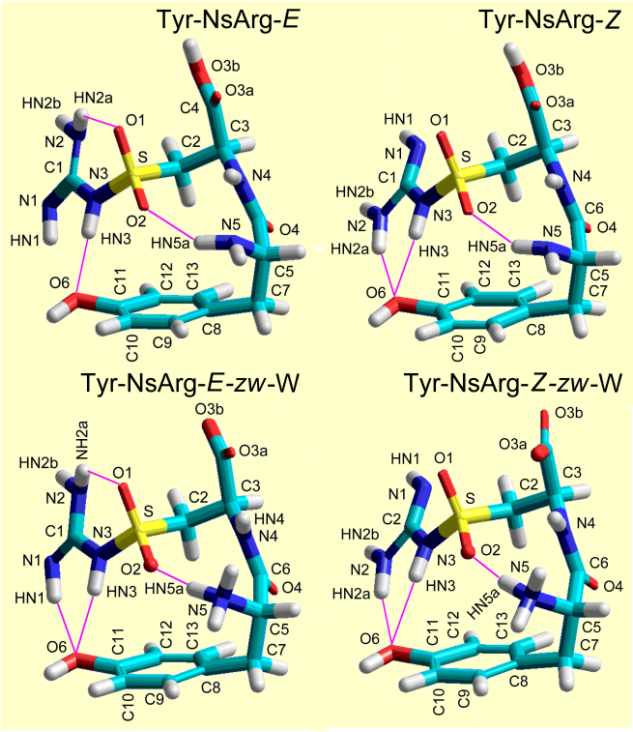


Fig. 3. HF *ab initio* optimized LECs for the neutral (Tyr-NsArg-*E* and Tyr-NsArg-*Z*) and *zwitter*-ionic (Tyr- NsArg-*E*-*zw*-W and Tyr- NsArg-*Z*-*zw*-W) forms of tyrosine-norsulfoarginine*E*- and *Z*-diastereomers in gas phase and water solution (W) with atom labeling and H-bonds

In all the three cases, H-bonding plays an important role (Figs. 1-3; the close contacts of the type H···A are marked with lines). For kyo, the guanidino group, the phenolic hydroxyl and the carboxylic group are involved in H-bonding. Additionally, the amino group and the peptide oxygen participate in weaker H-bonds. In the case of NsArg-Tyr and Tyr-NsArg, the guanidino group forms H-bonds with the phenolic hydroxyl and one or the two supfonamide oxygen atoms. Special attention deserve the

very short (*ca*. 1.75 Å) H-bonds of the type N-H···O which link the protonated amino group and one of the sufo-oxygen atoms in the *zwitter* ionic forms Tyr-NsArg-*E*-*zw*-W and Tyr-NsArg-*Z*-*zw*-W (Table 3).

Table 2. Energy differences (*Δ*E) and selected geometric parameters for norsulfoarginine-tyrosine diastereomers in neutral (NsArg-Tyr) and *zwitter*-ionic (NsArg-Tyr-zw) form in gas phase and in water medium (See Fig. 2)

Geometric parameter	Molecule					
	NsArg-Tyr-*E*[a]	NsArg-Tyr-*E*-W	NsArg-Tyr-*Z*	NsArg-Tyr-*Z*-W	NsArg-Tyr-*E*-*zw*-W	NsArg-Tyr-*Z*-*zw*-W
			*Δ*E, Hartree[b]			
	0.0517[b]	0.0019	0.0542	0.0000	0.0502	0.0539
			Bond lengths, Å			
N1-C1	1.259	1.263	1.257	1.265	1.261	1.262
N2-C1	1.376	1.370	1.369	1.359	1.371	1.358
C1-N3	1.418	1.418	1.425	1.424	1.425	1.434
N3-S	1.627	1.622	1.620	1.616	1.610	1.604
SC2	1.759	1.750	1.761	1.752	1.765	1.769
O6-C11	1.400	1.399	1.412	1.409	1.399	1.404
C5-C6	1.511	1.512	1.512	1.512	1.548	1.548
C6-O4	1.202	1.208	1.203	1.208	1.247	1.247
C6-O5	1.357	1.344	1.355	1.343	1.255	1.254
			Sum of bond angles at C and N atoms of guanidine fragment, deg			
Σang(C1)	359.9	359.9	360.0	360.0	359.9	360.0
Σang(N2)	351.6	353.7	359.1	358.5	351.6	360.0
Σang(N3)	359.4	359.0	359.4	358.7	359.2	360.0
			Torsion angles, deg			
C1-N3-S-C2	-70.0	-68.4	-70.9	-70.5	-69.2	-81.4
C9-C8-C7-C5	-65.1	-65.1	-58.2	-58.6	-70.6	-68.5
O6-C11-C10-C9	-173.8	-173.1	-174.3	-172.9	-173.0	-170.9
C10-C9-C-8-C7	173.1	173.3	175.4	175.6	175.4	168.2
			H-bonding geometry			
N3-HN3	1.001	1.006	0.996	1.002	1.002	1.001
HN3···O6	1.951	1.904	2.009	1.913	1.948	1.940
N3-HN3···O6	162.7	163.0	150.3	153.2	149.7	157.9
N2-HN2a	0.992	0.991	0.988	0.990	0.992	0.990
HN2a···O6			2.307	2.246		2.158
N2-HN2a···O6			145.6	144.8		148.3
HN2a···O1	2.065	2.070			2.125	
N2-HN2a···O1	129.6	129.8			126.9	
N1-HN1	0.997	0.998	0.996	0.997	0.999	0.995
HN1···O1			2.238	2.171		2.193
N1-HN1-O1			131.9	134.4		132.7
HN1···O6					2.335	
N1-HN1···O6					143.8	

[a] *E* and *Z* denote the configuration of the N1=C1 bond with respect to N3-S bond.
[b] The absolute energies are: -1613.7592, -1613.8090, -1613.7567, -1613.8109, -1613.7611 and -1613.7570 Hartree, respectively.

Table 3. Energy differences (*Δ*E) and selected geometric parameters for tyrosine-norsulfoarginine diastereomers in neutral (Tyr-NsArg) and *zwitter*-ionic (Tyr-NsArg-zw) form in gas phase and in water medium (See Fig. 3)

Geometric parameter	Molecule					
	Tyr-NsArg-*E*[a]	Tyr-NsArg-*E*-W	Tyr-NsArg-*Z*	Tyr-NsArg-*Z*-W	Tyr-NsArg-*E*-*zw*-W	Tyr-NsArg-*Z*-*zw*-W
			*Δ*E, Hartree[b]			
	0.0397	0.0036	0.0411	0.0000	0.0331	0.0284
			Bond lengths, Å			
N1-C1	1.256	1.261	1.262	1.266	1.260	1.266
N2-C1	1.377	1.370	1.368	1.358	1.371	1.357
C1-N3	1.423	1.423	1.403	1.410	1.422	1.410
N3-S	1.620	1.617	1.630	1.620	1.620	1.622
SC2	1.758	1.757	1.757	1.756	1.758	1.760
O6-C11	1.401	1.397	1.404	1.397	1.391	1.391
C3-C4	1.508	1.509	1.508	1.509	1.556	1.558
C4-O3a	1.198	1.203	1.199	1.203	1.253	1.256
C4-O3b	1.351	1.342	1.349	1.343	1.243	1.241
			Sum of bond angles at C and N atoms of guanidine fragment, deg			
Σang(C1)	360.0	360.0	360.0	360.0	360.0	360.0
Σang(N2)	351.3	352.8	359.2	359.3	352.0	359.3
Σang(N3)	359.8	359.7	360.0	350.9	359.9	360.0
			Torsion angles, deg			
C1-N3-S-C2	-65.3	-65.1	-56.8	-56.3	-66.5	-59.9
C9-C8-C7-C5	-84.0	-85.3	-84.2	-85.6	-90.3	-90.7
O6-C11-C10-C9	-169.9	-170.6	-171.4	-170.9	-170.0	-172.6
C10-C9-C-8-C7	169.9	169.9	171.9	170.4	170.9	172.7
			H-bonding geometry			
N1-HN1	0.998	0.999	0.996	0.997	0.999	0.997
HN1···O6					2.360	
N1-HN1···O6					152.4	
N3-HN3	0.996	0.999	0.991	0.996	0.994	0.992
HN3···O6	2.063	2.021	2.269	2.120	2.201	2.465
N3-HN3···O6	158.4	155.7	147.6	147.3	152.5	141.1
N2-HN2a	0.992	0.991	0.992	0.992	0.991	0.994
HN2a···O1	2.123	2.140			2.185	
N2-HN2a···O1	127.9	127.9			127.6	
HN2a···O6			2.117	2.116		2.062
N2-HN2a···O6			156.7	152.2		162.5
N5-HN5a	0.997	0.997	0.997	0.997	1.023	1.025
HN5a···O2	2.308	2.585	2.184	2.545	1.749	1.731
N5-HN5a···O2	145.8	134.3	148.0	135.0	169.5	171.6

[a] *E* and *Z* denote the configuration of the N1=C1 bond with respect to N3-S bond.

[b] The absolute energies are: -1613.7843, -1613.8204, -1613.7829, -1613.8240, -1613.7909 and -1613.7956 Hartree, respectively.

Acknowledgments. The work is financially supported by Bulgarian Ministry of Higher Education and Science (Grants MY-FS-13/07 and DNS-7FP-02/1) and by the European FP7 HP-SEE Project (Contract No RI-261499).

References

1. Fukui, K., Shiomi, H., Takagi, H., Hayashi, K., Kiso, Y., Kitagawa, K.: Isolation from Bovine Brain of a Novel Analgesic Pentapeptide, Neo-kyotorphin, Containing the Tyr-Arg (Kyotorphin) Unit. Neuropharmacology 22, 191–196 (1983)
2. Amano, H., Morimoto, Y., Kaneko, S., Takagi, H.: Opioid Activity of Enkephalin Analogues Containing the Kyotorphin-related Structure in the N-terminus. Neuropharmacology 23, 395–400 (1984)
3. Ueda, H., Ming, G., Hazato, T., Katayama, T., Takagi, H.: Degradation of Kyotoprhin by a Purified Membrane-bound Aminopeptidase from Monkey Brain: Potentiation of Kyotorphin-induced Analgesia by a Highly Effective Inhibitor. Bestatin. Life Sci. 36, 1865–1871 (1985)
4. Dzimbova, T., Pajpanova, T., Golovinsky, E.: Collect. Czech. Chem. Commun.: Synthesis of Some Sulfoguanidino Group-Containing Amino Acids 6, 12–14 (2003)
5. Dzimbova, T., Pajpanova, T., Golovinsky, E.: Synthesis of Novel Kyotorphins Containing Non-proteinogenic Amino Acid norsulfoarginine. Bulg. Chem. Commun. 38, 45–48 (2006)
6. Bocheva, A., Dzambazova-Maximova, E., Dzimbova, T., Pajpanova, T., Golovinsky, E.: Analgesisc Action of NorsulfoarginineKyotorphin Analogue in Rats. Compt. Rend. Acad. Bulg. Sci. 59, 219–222 (2006)
7. Hypecube, Inc., Hyperchem. Release 7.52 (2002)
8. EMSL Basis Set Exchange Library, `https://bse.pnl.gov/bse/portal`
9. Granovsky, A.A.: Firefly Version 7.1.G (2009), `http://classic.chem.msu.su/gran/gamess/index.html`
10. GAMESS Version 25 MAR 2010 (R2); Schmidt, M.W., Baldridge, K.K., Boatz, J.A., Elbert, S.T., Gordon, M.S., Jensen, J.H., Koseki, S., Matsunaga, N., Nguyen, K.A., Su, S., Windus, T.L., Dupuis, M., Montgomery, J.A.: General Atomic and Molecular Electronic Structure System. J. Comput. Chem. 14, 1347 (1993)

Dynamics of Uninhibited and Covalently Inhibited Cysteine Protease on Non-physiological pH

Branko J. Drakulić[1,*] and Marija Gavrović-Jankulović[2]

[1] Department of Chemistry-IChTM, University of Belgrade,
Njegoševa 12, 11000 Belgrade, Serbia
bdrakuli@chem.bg.ac.rs
[2] Faculty of Chemistry, University of Belgrade,
Studentski trg 16, 11000 Belgrade, Serbia
mgavrov@chem.bg.ac.rs

Abstract. Differences in activity and the structural stability under simulated gastric juice conditions of uninhibited and covalently inhibited cysteine protease, isolated from the fruit, were experimentally observed. We employed molecular dynamics simulations of proteins modeled from the similar ones with known 3D structure to explain experimental findings. Simulations were performed with NAMD, using CHARMM force field in explicit solvent model. Conformational changes observed in MD trajectories offer indication on differences in stability of inhibited *vs*. uninhibited protein on low pH values. Protonation states of the protein side chains, through the non-bonded interactions that stabilize 3D structures, likely, significantly contribute to difference in stability of uninhibited and covalently inhibited protein on low pH values.

Keywords: Cysteine-protease, Molecular dynamics, Protonation states.

1 Introduction

Cysteine proteases, involved in degradation of proteins, widespread in plants, parasites and vertebrates, are an important medicinal chemistry target implicated in the diseases ranging from infections, immunological processes to cancer [1, 2]. Due to conserved cysteine residue in their active site, compounds comprising activated double bonds (-CH=CH-EWG (electron withdrawn group), where EWG is -C(O)-, -C(O)NH-, -CN, $-NO_2$, $-SO_2-$ …) and their analogs (mainly oxiranes and aziridines bound to EWG), represent the major chemotypes used for inhibitors design [3]. Typically, enzyme active site comprise of catalytic dyad CYS-HIS, in which proximal HIS deprotonates CYS -SH group and facilitate catalysis, *i.e.* cleavage of the peptide bond. Covalent inhibitors forms covalent bond with the -SH group of active site CYS, by Michael-type addition. The E-64 (Fig. 1) is common cysteine proteases inhibitor isolated from mold *Aspergillus japonicus*.

* Corresponding author.

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_9, © Springer International Publishing Switzerland 2014

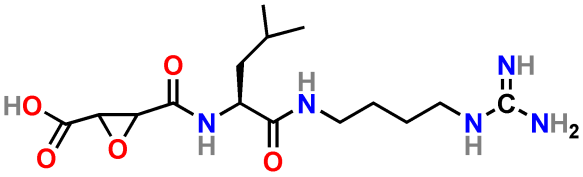


Fig. 1. Structure of E-64

In Protein Data Bank [4] fifteen structures of cysteine protease covalently inhibited with E-64 can be found. Cysteine proteases from papain family, mainly found in fruits, are known allergens. Actinidin is a cysteine protease from kiwifruit. Kiwifruit is ranked in the top-ten food allergen sources [5]. Protein is composed of two domains divided by the cleft (Fig. 2), in which natural substrates was bound prior to catalysis. The active site dyad CYS25-HIS162 is situated on the end of this cleft. Three disulfide bridges (two in domain II, and one in domain I) stabilize tertiary structure of the protein.

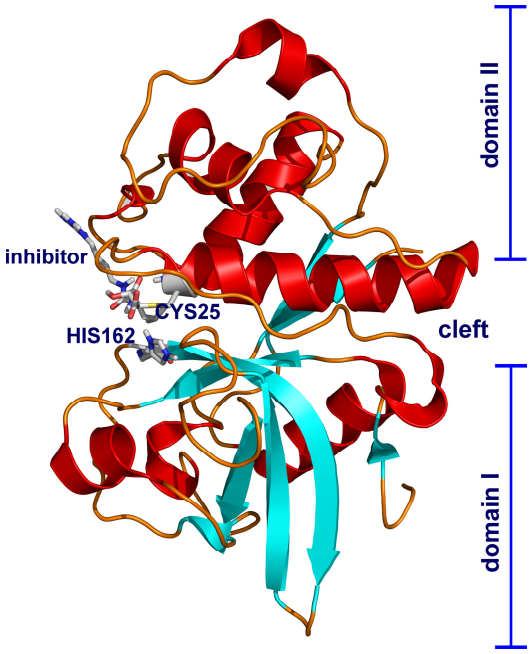


Fig. 2. Structure of actinidin (depicted from PDB entry 1AEC)

Protonation states of amino-acid side-chains bearing ionizable functional groups strongly depend on pH and on local environment. On physiological pH (7.04) aspartic and glutamic acids are deprotonated (have negative charge), arginine and lysine are protonated (have positive charge), while histidine is neutral. On low pH (as in gastric

juice, pH ~ 1.5), aspartic and glutamic acids are neutral, while arginine, lysine and histidine are protonated (have positive charge). Due to more acidic than basic amino acid residues in its sequence, actinidin isolated from kiwifruit, that is subject of this report, on physiological pH bear net-negative charge. Switching to gastric juice pH protein attain net-positive charge. This significantly influences non-bonded interactions that stabilize 3D structure of the protein, because the salt bridges (form of the strong non-covalent interactions) are changed to hydrogen bonds (which are something weaker), as will be shown in following text.

2 Results and Discussion

We have experimentally observed that actinidin, isolated from kiwifruit, covalently inhibited with E-64 was more susceptible to proteolysis under simulated gastric juice conditions compared to uninhibited enzyme [6]. Spectral data (UV and CD spectra) suggested conformational change of protein upon inhibitor binding. We performed molecular dynamics (MD) simulation of inhibited and uninhibited protein during 5 ns in explicit solvent model. Inhibited protein was modeled from the PDB structure 1AEC [7], while uninhibited protein was modeled from PDB structure 2ACT [8]. Entries taken from the PDB are phylogenetically and structurally closely related to actinidin isolated from kiwifruit, for more details see Experimental section. Low pH value of medium has been simulated by assigning corresponding protonation states to ionizable amino-acid side chains. Protonation states were assigned using empirical function [9]. Even 5 ns is too short time to any unfolding of the protein be observed, see for example reference [10], data obtained from the analysis of MD trajectories gave indication on difference in stability of inhibited *vs*. uninhibited enzyme. We analyzed stability of tertiary structure of proteins, conformational changes, possible difference of the position of the hydrophobic residues that could become exposed to solvent and the movement of covalently bound ligand. On the first visual inspections of MD trajectories, inhibited protein appears something more compact than uninhibited one; judging by, for example, the radius of gyration (rgyr) of TYR side-chains. Modeled actinidin comprise 14 TYRs' exposed to solvent. The peak of rgyr for inhibited protein was at ~ 18 Å, while the peak of rgyr for uninhibited protein was at ~ 19 Å. Such data suggested opposite behavior comparing to experiments.

Afterward we analyzed stability of non-bonded intramolecular interactions that stabilize tertiary structure of proteins under study. Under physiological conditions (pH 7) two salt bridges, LYS17-GLU86 and LYS181-GLU35, stabilize the interdomain cleft (Fig. 3). Simulations performed on physiological pH will be described elsewhere [11]. During 5 ns of MD simulations at pH 7, we observed minor fluctuations of distances measured between LYS NZs' and GLU CDs' of these salt bridges, both in inhibited and in uninhibited enzyme (data not shown).

The analysis of MD trajectories of uninhibited enzyme on low pH reveled that distances between side-chain $-NH_3^+$ of LYS17 and LYS181 and corresponding side-chain -COOH groups of GLUP86 and GLUP35 became significantly longer

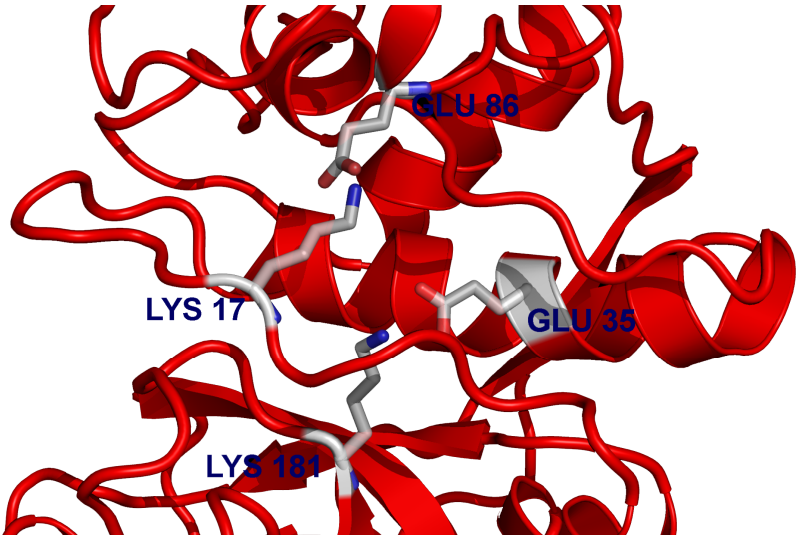


Fig. 3. Salt bridges that stabilize interdomain cleft under physiological pH

(Figure 4 a and b), comparing to simulation on physiological pH. As described above, salt bridges that exist in the enzyme on physiological pH, as the stronger type of electrostatic non-bonded interactions, were turned to hydrogen bonds. Interdomain cleft became wider, and in uninhibited enzyme such widening has been symmetrical along the whole cleft (see Fig. 7 a). Enzyme, likely, could accommodate its natural substrates, and retain proteolytic activity.

The analysis of MD trajectories of inhibited enzyme on low pH reveled different behavior. LYS17 and GLUP86 were close to each other during ~ 3.5 out of 5 ns of MD simulation. Distance between -$\underline{N}H_3^+$ of LYS17 and -$\underline{C}OOH$ of GLUP86 were ~ 3.5 Å (Fig. 4 c), so such nonbonded interaction stabilize part of the cleft near to active site dyad (CYS25 - HIS162). On the other hand, distance between -NH_3^+ of LYS181 and -COOH of GLUP35 became even longer than in uninhibited enzyme (Fig. 4 d). We observed additional stabilization of the cleft, near the end on which the catalytic dyad was situated (Fig. 5). LYS17, along with GLUP86, forms hydrogen bond with GLUP50 (Fig. 6 a). Such non-bonded interaction cannot be observed in uninhibited protein (Fig. 6 b). One more hydrogen bond pair that stabilize interdomain cleft near the catalytic dyad was also observed. Inhibitor is covalently bound to CYS25 that belong to domain II. HIS162 is part of domain I (see Figure 1). The imidazolyl ring of HSP162 and -COOH group of the E64 form hydrogen bond (Fig. 5). Those two moieties were close to each other during whole 5 ns of MD simulations of inhibited enzyme (Fig. 6 c).

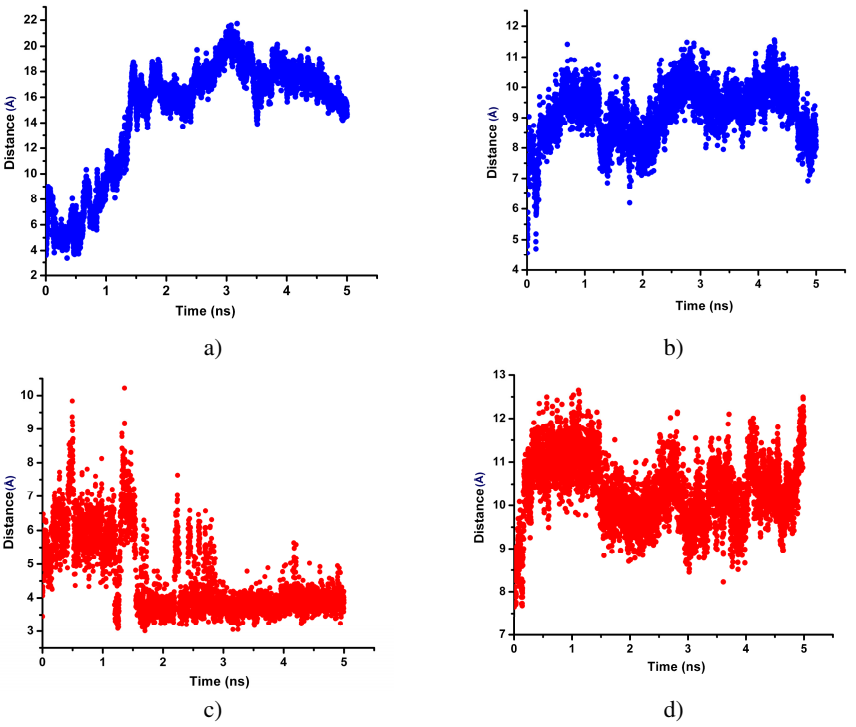


Fig. 4. Distance between LYS17 and GLUP86 of uninhibited (a), and inhibited enzyme (c). Distance between LYS181 and GLUP35 of uninhibited (b), and inhibited enzyme (d). Measured as –**N**H_3 to –**C**OOH distance.

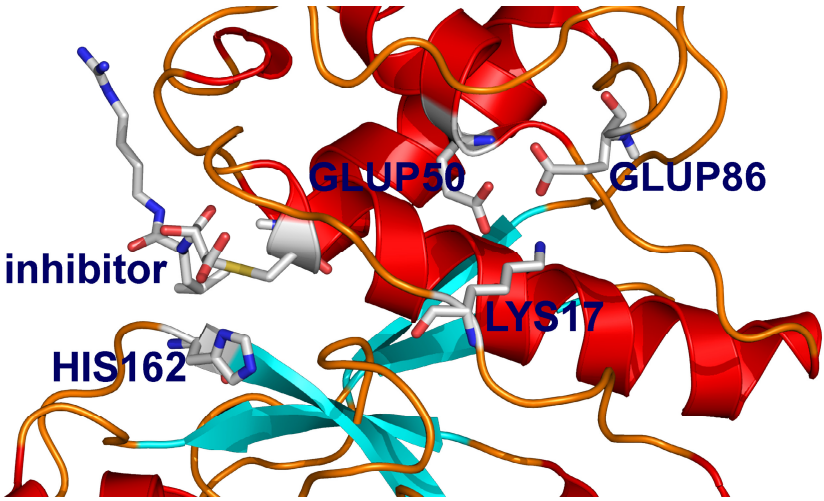


Fig. 5. Interactions that additionally stabilize interdomain cleft near the catalytic dyad of inhibited enzyme

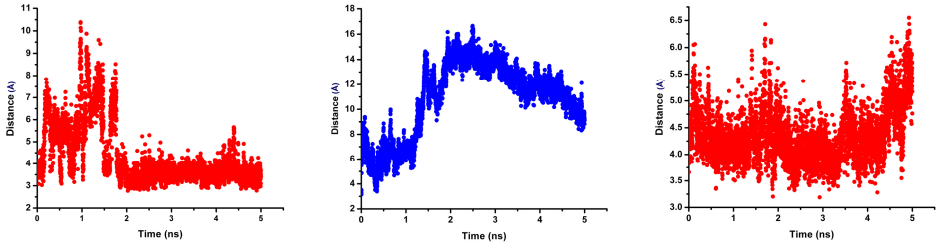


Fig. 6. Distance between LYS17 and GLUP50 of inhibited (a), and uninhibited enzyme (b), measured as –**N**H_3 to –**C**OOH distance. Distance between E-64 and HSP162 (**c**), measured as distance of –**C**OOH to centroid of imidazolyl ring of HSP162.

Simultaneous H-bonding of the LYS17 with GLUP86 and GLUP50, together with HSP162 to E64 H-bonding, influences non-symmetrical widening of the interdomian cleft in inhibited protein, as is shown on Fig. 7 b. Such destabilization of the tertiary structure of the protein, probably, by time destroys integrity of the inhibited protein in extent sufficient to be easily attacked by gastric juice proteases. Differences in conformations of the inhibited and uninhibited protein, obtained after 5 ns of MD simulations are schematically depicted on the Fig. 7.

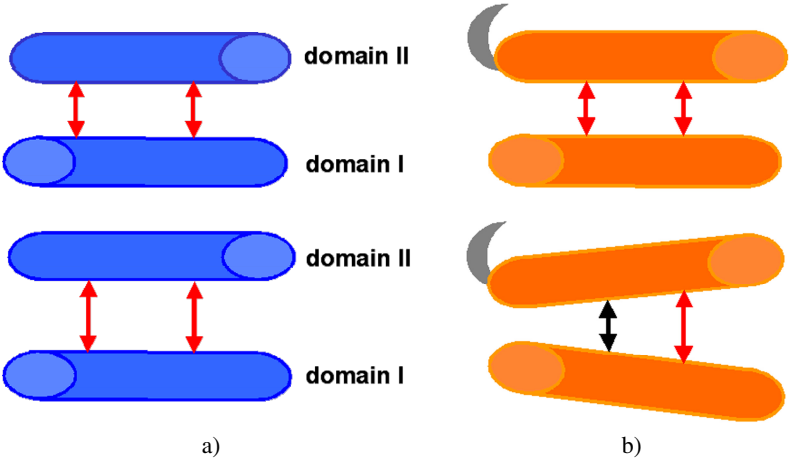


Fig. 7. Schematic representation of the changes of interdomain cleft during MD simulations. Uninhibited enzyme (a), inhibited enzyme (b). Up – equilibrated system, down – system after 5 ns of MD simulations. Inhibitor is schematically shown in gray.

3 Conclusions

Following experimental observation of differences in proteolytic activity and structural integrity under simulated gastric juice conditions of uninhibited and covalently inhibited actinidin, cysteine protease isolated from kiwifruit, we performed MD simulations of modeled enzymes. Along with biological activity, uninhibited and inhibited

enzymes differ in structural stability in the simulated gastric conditions. Results of MD simulations suggested that changes of protonation states of ionizable amino-acid side chains under low pH influences non-covalent interactions that stabilize tertiary structure of the enzyme. Two salt bridges that exist on physiological pH stabilize interdomain cleft. On low pH values those salt bridges turn to hydrogen bonds. We observed symmetrical widening of the interdomain cleft in the uninhibited enzyme, which probably still allow fitting of natural substrates, and retaining of the proteolytic activity of the enyzme. In inhibited enzyme widening of the interdomain cleft were highly unsymmetrical. This probably cause losing of the tertiary structure of the enzyme, and allow proteolytic digestion of the enzyme.

4 Experimental

Inhibited enzyme has been modeled from PDB entry 1AEC, the actinidin protein isolated from *Actinidia chinesis* solved on 1.9 Å resolution. Uninhibited enzyme has been modeled from PDB entry 2ACT, the actinidin protein isolated from *Actinidia chinesis* solved on 1.7 Å resolution. Sequence of the actinidin on which experimental data were collected was aligned with sequence of PDB entries in Clustal X [12]. Amino acid side chains that differed real protein from templates were manually changed in VegaZZ 2.0.4 [13], as well as corresponding amino-acid names. Afterward we checked any eventual close contact or alike. The pK_a values of ionizable side-chains on pH 1 were ascribed by Propka 2.0 [9]. Type of amino acids were changed, ASP to ASPP, GLU to GLUP, HIS to HSP, to fit to CHARMM definition of amino-acids protonated on low pH values, and hydrogens were added to -COO groups of GLUP and ASPP. Both N of the imidazolyl ring in HSP was also protonated. Each system under study was neutralized by addition of counter ions, and afterward embedded in explicit water cluster (90 Å). CHARMm 22 force filed was used, with Gasteiger charges. All MD simulations were performed in NAMD 2.8 [14], on Linux cluster equipped with 2 x quad core Intel Xeon-E5345 @ 2.33 GHz processors. Each system was minimized during 30 000 steps, than heated to 300 K during 10 000 steps. The 5 ns of unconstrained, productive, molecular dynamics simulations were performed for each system, at 300 ± 10 K (Langevin's algorithm), using periodic boundary conditions. Electrostatics was treated by a Particle Mesh Ewald algorithm. Cut-offs were set to 12 Å for vdW interactions, with a switch starting at 8 Å, and pair list distance set to 13.5 Å. Trajectory frames were collected each ps to yield 5000 frames per trajectory. The each simulation was repeated three times using different random seeds, giving very comparable results. The VegaZZ were used for the analysis of MD trajectories. Figures were made by PyMol 0.99 [15].

Acknowledgements. The work reported makes use of results produced by the High-Performance Computing Infrastructure for South East Europe's Research Communities (HP-SEE), a project co-funded by the European Commission (under Contract Number 261499) through the Seventh Framework Programme HP-SEE (`http://www.hp-see.eu/`). Ministry of Education, Science and Technological Development of Serbia support this work under grant 172035.

References

1. Chapman, H.A., Riese, R.J., Shi, G.P.: Emerging roles for cysteine proteases in human biology. Annu. Rev. Physiol. 59, 63–68 (1997)
2. Sajid, M., McKerrow, J.H.: Cysteine proteases of parasitic organisms. Mol. Biochem. Parasitol. 120, 1–21 (2002)
3. Santos, M.M.M., Moreira, R.: Michael acceptors as cysteine protease Inhibitors. Mini-Rev. Med. Chem. 7, 1040–1050 (2007)
4. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The Protein Data Bank. Nucl. Acids Res. 28, 235–242 (2002), http://www.rcsb.org
5. Bublin, M., Pfister, M., Radauer, C., Oberhuber, C., Bulley, S., Marknell-DeWitt, A., Lidholm, J., Reese, G., Vieths, S., Breiteneder, H., Hoffmann-Sommergruber, K., Ballmer-Weber, B.K.: Component-resolved diagnosis of kiwifruit allergy with purified natural and recombinant kiwifruit allergens. J. Allergy Clin. Immunol. 125, 687–694 (2010)
6. Grozdanovic, M.M., Aleksic, I., Burazer, L., Andjelkovic, U., Petersen, A., Gavrovic-Jankulovic, M.: Actinidin, a specific biomarker of kiwifruit allergy, exerts proteolytic activity upon treatment in the simulated gastrointestinal environment (2012) (submitted)
7. Varughese, K.I., Su, Y., Cromwell, D., Hasnain, S., Xuong, N.H.: Crystal structure of an actinidin-E-64 complex. Biochemistry 31, 5172–5176 (1992)
8. Baker, E.N., Dodson, E.J.: Crystallographic refinement of the structure of actinidin at 1.7 angstroms resolution by fast Fourier least-squares methods. Acta Crystallogr., Sect. A 36, 559 (1980)
9. Bas, D.C., Rogers, D.M., Jensen, J.H.: Very fast prediction and rationalization of pKa values for protein-ligand complexes. Proteins 73, 765–783 (2008), http://propka.ki.ku.dk/
10. Eberini, I., Emerson, A., Sensi, C., Ragona, L., Ricchiuto, P., Pedretti, A., Gianazza, E., Tramontano, A.: Simulation of urea-induced protein unfolding: A lesson from bovine β-lactoglobulin. J. Mol. Graph. Mod. 30, 24–30 (2011)
11. Grozdanović, M., Drakulić, B.J., Gavrović-Jankulović, M.: Conformational mobility of active and E-64-inhibited actinidin (submitted 2013)
12. Thompson, J.D., Gibson, T.J., Plewniak, F., Jeanmougin, F., Higgins, D.G.: The Clus-talX windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. Nucl. Acids Res. 25, 4876–4882 (1997)
13. Pedretti, A., Villa, L., Vistoli, G.: Vega - An open platform to develop chemo-bioinformatics applications, using plug-in architecture and script programming. J. Comp. Aided Mol. Des. 18, 167–173 (2004), http://www.ddl.unimi.it
14. Phillips, J.C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., Chipot, C., Skeel, R.D., Kalé, L., Schulten, K.: Scalable molecular dynamics with NAMD. J. Compt. Chem. 26, 1781–1802 (2005), http://www.ks.uiuc.edu/Research/namd/
15. The PyMOL Molecular Graphics System, Version 0.99, Schrödinger, LLC, http://www.pymol.org

Mechanisms of Polarization

Aggelos Avramopoulos, Heribert Reis, and Manthos G. Papadopoulos

Institute of Biology, Medicinal Chemistry and Biotechnolodgy,
National Hellenic Research Foundation,
48 Vas. Constadinou Ave., Athens, 11635, Greece
{aavram,hreis,mpapad}@eie.gr

1 Introduction

For several years we have been working on the computation of the linear and non-linear optical (L&NLO) properties of organic molecules and on the design/selection of derivatives, which may have applications in the development of photonic devices. There is an intensive effort all over the world to discover the mechanisms and factors, which lead to large NLO properties, which are some of the properties required for materials to be useful in photonic applications. Here we review some of these factors, which lead to interesting or very large NLO properties.

2 Methods

When we set a molecule in a uniform electric field, F, its energy, E(F), is expanded in a Taylor series:

$$E(F)= E^0 - \mu_i F_i - (1/2)\alpha_{ij}F_iF_j - (1/6)\beta_{ijk}F_iF_jF_k - (1/24)\gamma_{ijkl}F_iF_jF_kF_l -..., \quad (1)$$

where E^0 is the field independent energy of the molecular system, F_i, F_j, F_k,F_l are the field components μ_i, α_{ij}, β_{ijk} and γ_{ijkl} are the components of the dipole moment, polarizability, first and second hyperpolarizability, respectively.

The average polarizability and second hyperpolarizability we consider in this review are given by:

$$\alpha = \frac{1}{3}\sum_{i=x,y,z}\alpha_{ii} \quad (2)$$

$$\gamma = \frac{1}{5}\sum_{ij=x,y,z}\gamma_{iijj} \quad (3)$$

The (hyper)polarizabilities have been computed by employing a wide variety of methods involving semi-empirical and ab-initio techniques. Earlier studies have been performed by employing low-level *ab-initio* methods or semi-empirical techniques (e.g. Austin Model 1-AM1). In recent years rigorous methods (e.g. coupled cluster

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_10, © Springer International Publishing Switzerland 2014

with iterative computation of single and double excitations and perturbative treatment of triple excitations -CCSD(T)-, complete active space SCF/CAS second-order perturbation theory -CASSCF/CASPT2-) have been employed, in particular for small molecules (e.g. AuXeF, HXe_2F). A detailed account of the approaches which have been used is given in the original articles we review. In this article we shall only consider the electronic contributions to L&NLO properties.

Computer resources: Over the years we have used several computer facilities, among which we note the TeraGrid, Barcelona supercomputer centre (BSC), PRACE, High Performance Computing for South East Europe (HP-SEE), besides our local computing infrastructure.

3 Results and Discussion

3.1 Squaraines

The linear and non linear optical properties (L&NLO) of a monolayer of (4-(N-methyl-N-(carboxypropyl)amino)-phenyl-4′-(N,N-dibutylamino)phenylsquaraine (SBA) were computed by a combination of molecular dynamics simulations, molecular quantum mechanical calculations and a discrete local field model to compute the macroscopic optical susceptibilities [1]. In addition, SBA in water and dimethylsulfoxide (DMSO) solutions were simulated and the aggregation behavior of SBA in these solvents was analysed.

The electronic structure and intermolecular interactions of squaraine dyes have been an area of intense interest for more than three decades. Squaraines can be described as quadrupolar molecules, which are highly polarizable and thus may exhibit large and unusual linear and nonlinear optical (L&NLO) properties. Apart from interesting molecular properties, such as negative second hyperpolarizabilities [2,3] and negative electric-field induced second harmonic generation signals [4,5], there are strong intermolecular interactions between squaraine molecules in larger aggregates which can lead to quite unusal L&NLO properties. For example, Ashwell et al. have observed strong second-harmonic generation (SHG) signals from Langmuir-Blodgett films composed of essentially centrosymmetric squaraine dyes [6]. We use classical molecular dynamics to simulate thin films of a squaraine dye and perform quantum-chemical calculations for the characterization of the molecular (hyper)polarizabilities. We also extended a discrete local field model [7] in order to be able to predict the L&NLO susceptibilities of Langmuir-Blodgett films and applied this model to the simulated LB films. We chose to use 4-(N-methyl-N-(carboxypropyl)amino)-phenyl-4′-(N,N-dibutylamino)phenylsquaraine (SBA, Figure 1) as our model system, which has been investigated extensively both in solution and in Langmuir and Langmuir-Blodgett films by Chen et al. [8,9]. Their results can thus be used as a valuable reference for the theoretical predictions, in particular for the simulations of the macroscopic assemblies. In the first step, molecular simulations were performed with the GROMACS molecular dynamics (MD) package, version 4.0.3. For the Langmuir film, the simulation box consisted of two SBA interfaces, with a layer of water in between. A monolayer of SBA was also simulated on a solid surface (graphite).

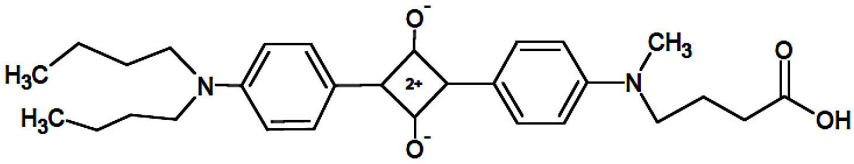


Fig. 1. Schematic representation of a SBA molecule

As a second ingredient of the approach, the molecular static (hyper)polarizabilities of SBA were computed at the DFT as well as MP2 level of theory, using the Gaussian 09 program package. For the DFT calculations, the B3LYP functional combined with aug-cc-pVDZ and 6-31+G** basis sets were employed. In order to validate the DFT results, a MP2/6-31+G** calculation was performed for selected (hyper)polarizability components. Agreement between MP2 and DFT values was satisfactory. In addition to the electric properties, the excitation spectrum was computed both for a monomer and a dimer using TDDFT, with B3LYP/6-31+G**, both in Gas phase as well as in water and in dimethylsulfoxid (DMSO), applying the polarizable continuum model (PCM). For the optimized dimer with *parallel* dipole moments (i.e. a conventional H-dimer) in water or DMSO solution, a small blue-shift of ~5 nm was obtained in comparison to the monomer, much smaller than the experimentally observed shift (~56 nm).

Finally, both the simulated film trajectories were analyzed using the so-called discrete local field approximation. In this model, the local field is computed in the dipole approximation and the effect of the local field on the (hyper)polarizabilities is then taken into account by a new round of quantum-chemical calculations. Finally, the macroscopic susceptibilities are computed using the trajectories. More details of this method are given in Ref. 7. The Langmuir system investigated here suffered from the 'polarization catastrophe' problem. In order to circumvent this problem, we introduced an adaptation of the “smearing” method previously developed by Thole [10] into the discrete local field method.

The average local field obtained for SBA in the films were quite small, as expected. The most interesting feature of the computed nonlinear susceptibilities is a large anisotropy of the first nonlinear susceptibilty in the surface plane; in fact, the larger of the two in-plane components is even larger than the out-of-plane component, which is the one accessible to measurement. The in-plane anisotropy was interpreted as being more consistent with a perpendicular glide-layer packing than with a translational layer packing. The simulated nonlinear susceptibilities of the monolayer on graphite were found to be remarkably similar to those of the layers on water.

3.2 Lithiated Derivatives

Two functionals have been employed for the computation of the average polarizability, α, and second hyperpolarizability, γ of $Si_{20}H_{20}$, $C_{20}H_{20}$, $Si_{20}Li_{20}$ [11] : B3LYP and CAM-B3LYP. Both functionals give results (Table 1) which agree qualitatively

with each other. B3LYP and CAM-B3LYP give α and γ for $C_{20}H_{20}$ which are in satisfactory agreement with each other. Replacement of C with Si leads to a large increase in both α, and γ. Even a larger increase is observed by substituting H by Li. For example, $\gamma(Si_{20}Li_{20})/\ \gamma(Si_{20}H_{20}) = 74$. The effect of lithium and the alkali metals in general, on the NLO properties is well documented [13].

Table 1. The polarizability (α) and second hyperpolarizability (γ) of $C_{20}H_{20}$, $Si_{20}H_{20}$ and $Si_{20}Li_{20}$. The properties are reported in a.u. Basis Set: 6-31+G(2d) [11]

	α	γ
$Si_{20}Li_{20}$ [D_{5d}]		
B3LYP	1782	18.7×10^6
CAM-B3LYP	1580	10.1×10^6
$Si_{20}H_{20}$ [I_h]		
B3LYP	559.4	17.3×10^4
CAM-B3LYP	545.4	13.7×10^4
$C_{20}H_{20}$ [I_h]		
B3LYP	177.1	3.7×10^4
CAM-B3LYP	173.8	2.9×10^4

Table 2. Static polarizabilities and second hyperpolarizabilities (in a.u.) of hexalithiobenzene, calculated in different approximations [17]

	Method and Geometry[a]	
Property[b]	P/SCF// P/MP2	P/MP4[SDTQ]// P/MP2
α_{xx}	283.5	468.4
α_{zz}	115.3	152.4
α	227.4	363.1
γ_{xxxx} x 10^{-4}	193	913
γ_{zzzz} x 10^{-4}	18	57
γ_{xxzz} x 10^{-4}	23	92
γ x 10^{-4}	125	574

[a] P/M1//P/M2: property value calculated with basis P by using method M1 at molecular geometry computed with basis P and method M2. Basis P is the 6-31*+ set with orbital exponents of diffuse and polarization functions taken from ref. [20].

An early study on organolithium and organosodium compounds, employing specific examples $C_8H_6Li_2$ and $C_8H_6Na_2$ [12], has shown that $\gamma(C_8H_6Li_2)/\ \gamma(C_8H_6H_2)$ = 4.2 and $\gamma(C_8H_6Na_2)/\ \gamma(C_8H_6H_2)$ =16.7. A variety of organolithium derivatives (e.g. C_6Li_6) has been studied by using semi-empirical methods (MNDO and PM3). The great effect of lithiation on the L&NLO properties has been demonstrated [14]. Theologitis et al. [15] studied the polarizability and second hyperpolarizability of tetrakis(phenylethynyl)ethenes and several of its lithiated derivatives. The computations have been performed by employing the PM3 method [16]. The significant effect of lithiation, at a semi-empirical level of theory is demonstrated. Raptis et al.[17] computed the L&NLO properties of C_6Li_6. This molecule was synthesized by Simp et al. [18] and isolated by Baran et al. [19]. In Table 2 we present the polarizability and second hyperpolarizability. The properties have been computed at the SCF and MP4[SDTQ] levels of theory. Electron correlation greatly affects both the polarizability and the second hyperpolarizability components. For example, one observes: γ_{xxxx}(MP4[SDTQ])/ γ_{xxxx}(SCF) = 4.8.

3.3 Noble Gas Derivatives

We have demonstrated that insertion of a noble gas atom in the chemical bond A-B may lead to a very large increase of the NLO properties [21]. Specific examples we have considered are the insertion of Ar in HF leading to HArF and Xe into HC_2H leading to $HXeC_2H$ [22].

Several teams have inserted noble gas atoms in the bond A-B (e.g. H-C, H-O), leading to A-Ng-B, where Ng is the noble gas atom. In that way a large variety of compounds is produced [23-26]. These derivatives are metastable, the global minimum is Ng + HY. It has been found that the bonding in HNgY involves the structure HNg^+Y^-, where HNg^+ and Y^- are linked by a Coulombic interaction [27].

HArF

Kriachtchev et al. [28] reported the synthesis of HArF (argon fluorohydride). This is a linear molecule for which Wong et al. [29] found the following charge distribution: H(0.179), Ar(0.562), F(-0.741). These are Natural Bond Orbital (NBO) charges, which have been computed by employing the method QCISD/6-311++G**.

The results of Table 3 [30] show the properties of HArF computed by employing several basis sets and the methods SCF, MP2 and CCSD(T). It is observed that correlation has a significant effect on both α_{zz} and β_{zzz}. MP2 gives satisfactory results in comparison with those computed by the most accurate CCSD(T) method. Pol(65) gives results in satisfactory agreement with those derived by using the much larger aug-cc-pV5Z(338). In parenthesis the number of basis functions is given.

The inserted Ar atom has a great effect on the properties of HF (hydrogen fluoride), taking into account that its α_{zz} and β_{zzz} are 6.449 (5.59) and –10.11(-7.39) a.u., respectively. These properties have been calculated at the MP4[SDTQ] level.

In parenthesis the properties computed with the method HF/Pol are given, while at this level α_{zz} for Ar atom is 10.61 a.u.

The static first hyperpolarizability, $\beta(0;0,0)$ of HArF has been rationalized by invoking the two-state model [31]. It was found that the remarkable increase of β could be attributed to the significant lowering of the excited states associated with strong electronic transitions [21,30]. We have performed complete active space valence bond (CASVB) [32] calculations. The following significant resonance structures have been found from these computations: $H\text{-}Ar^{+}F(57\%) <-> H^{-}Ar^{2+}F^{-}$ (21%)<-> $H^{+}ArF^{-}$ (15%). The weight is given in parenthesis.

Table 3. The polarizability, α_{zz}, and first hyperpolarizability, β_{zzz} of HArF[a] [30]

	α_{zz}	β_{zzz}
	POL	
SCF	37.61	-597.8
MP2	55.37	-1220.9
CCSD(T)	59.80	-1418.1
	aug-cc-pVDZ	
SCF	37.34	-635.9
MP2	55.08	-1285.6
CCSD(T)	59.42	-1476.3
	aug-cc-pVTZ	
SCF	37.74	-608.2
MP2	54.82	-1197.5
CCSD(T)	57.93	-1324.2
	aug-cc-pVQZ	
SCF	37.80	-590.7
MP2	54.41	-1145.8
	aug-cc-pV5Z	
SCF	37.80	-578.7
MP2	54.01	-1102.5

[a] The computation of the properties has been performed by employing the geometry optimized, using the method CCSD(T)/aug-cc-pV5Z. The d and f orbitals involve 5 and 7 components, respectively.

AuXeF and XeAuF

We consider the (hyper)polarizabilities of AuXeF and XeAuF [33]. For comparison we have computed the properties of HXeF (Table 4). A series of compounds with formula NgMX, where Ng=Ar, Kr, Xe, M=Cu, Ag, Au and X=F, Cl, Br, have been synthesized [34-37]. AuXeF is a linear metastable compound. To the best of our knowledge HXeF has not yet been synthesized. HXeF is stable and its dissociation limit to neutral atoms is 2.4 eV [38]. A series of basis sets has been used to compute

α_{zz} and β_{zzz} of the molecules we study. The effect of basis set on α_{zz} is rather small. A bigger effect is observed on β_{zzz}. Replacement of H by Au leads to a significant increase of β_{zzz} (in absolute value). The position of Xe (AuXeF or XeAuF) has a significant effect on α_{zz} and even more on $\beta_{zzz.}$

Table 4. A basis set study of μ_z, α_{zz} and β_{zzz} of HXeF, AuXeF and XeAuF[a]. All values are in a.u. The relativistic correction has been computed by the Douglas-Kroll method [33].

Property / Method	α_{zz}			β_{zzz}		
	HXeF	AuXeF	XeAuF	HXeF	AuXeF	XeAuF
CCSD(T)						
B1[c]	61.01	188.06		-626	-1826	
B2[d]	60.95	186.68	77.14	-608	-1898	-265
B3[e]	58.08	190.70		-684	-1922	
B4[f]	59.22	187.37	73.05	-638	-2144	-150
B5[g]	59.88	185.73	76.11	-577	-2415	-266
B6[h]	59.70	184.26	75.98	-571	-2441	-265

[a] The molecules lie on the z-axis.

[b] Geometries were taken from Ref [39-41].

[c] Basis set: PolX_DK.

[d] Basis set: HyPolX_DK.

[e] Basis set: ANO-RCC-VTZP.

[f] Basis set: ANO-RCC-VTQP.

[g] Basis set: aug-cc-pVTZ. For Au and Xe an ECP was used. Core electrons are given in parenthesis: Au(79), Xe (28). [42,43].

[h] Basis set: aug-cc-pVQZ. For Au and Xe an ECP was used. Core electrons are given in parenthesis: Au(79), Xe (28). [42,43].

HXe_2F

Several derivatives involving the group Xe-Xe have been studied [44.45]. Frenking et al. [46] used quantum chemical methods to study the derivatives H-Ng-Ng-F, where Ng=Ar, Kr and Xe. Here, we briefly comment on the L&NLO properties of HXeF and HXe_2F. For comparison the properties of HF are also reported. From the results of Table 5 we observe the very large effect of Xe on the L&NLO properties:

$\beta_{zzz}(HXe_2F)/\ \beta_{zzz}(HXeF) = 19$, $\beta_{zzz}(HXeF)/\ \beta_{zzz}(HF) = 51$.

The effect on the second hyperpolarizability is even larger, e.g. $\gamma_{zzzz}(HXe_2F)/\ \gamma_{zzzz}(HF) = -14084$. All the ratios have been computed by employing values calculated at the CCSD(T)level of theory (Table 5).

Table 5. The (hyper)polarizabilities of HF, HXeF and HXe_2F[a] [21]

Property (au)	Method					
	HF			CCSD(T)		
	HF[b]	HXeF[c]	HXe_2F[c]	HF[b]	HXeF[c]	HXe_2F[c]
α_{zz}	5.59	51.25	206.96	6.19	59.59	420.43
β_{zzz}	-9.8	-436.7	-9633	-11.5	-582.1	-11040
(γ_{zzzz} x 10^{-3})	0.219	16.9	720	0.284	22.7	-4000

[a] The computations were performed with the aug-cc-pVDZ basis set. For Xe an effective core potential of 28 electrons was used [43].
[b] The experimental geometry was used [47].
[c] The geometries were taken from ref. 46.

Very significant is the effect of electron correlation on the (hyper)polarizabilities of HXe_2F and in particular the second hyperpolarizability: γ_{zzzz}(CCSD(T)/ γ_{zzzz}(HF) = -5.6.

Acknowledgement. Funding provided by the European Commission for FP7: a) REGPOT-2009-1, Project ARCADE. Grant Agreement No: 245866. b) Infrastructures -2010-2, HP-SEE. Grant Agreement No: 261499 is acknowledged, Computing resources by PRACE and HP-SEE are also acknowledged.

References

1. Megariotis, G., Avramopoulos, A., Papadopoulos, M.G., Reis, H.: J. Phys. Chem. C. 116, 15449 (2012)
2. Bigelow, R.W., Freund, H.-J.: Chem. Phys. 107, 159 (1986)
3. Dirk, C.W., Cheng, L.T., Kuzyk, M.G.: Int. J. Quant. Chem. 43, 27 (1992)
4. Chen, C.-T., Marder, S.R., Cheng, L.-T.: J. Chem. Soc., Chem. Commun. 259 (1994)
5. Chen, C.-T., Marder, S.R., Cheng, L.-T.: J. Am. Chem. Soc. 116, 3118 (1994)
6. Ashwell, G.J., Jefferies, G., Hamilton, D.G., Lynch, D.E., Roberts, M.P.S., Bahra, G.S., Brown, C.R.: Nature 375, 385 (1995)
7. Reis, H., Papadopoulos, M.G., Munn, R.W.: J. Chem. Phys. 1-9, 6828 (1998)
8. Chen, H., Herkstroeter, W.G., Perlstein, J., Law, K., Whitten, D.: J. Phys. Chem. 98, 5138 (1994)
9. Chen, H., Law, K., Perlstein, J., Whitten, D.G.: J. Am. Chem. Soc. 117, 7257 (1995)
10. Thole, B.T.: Chem. Phys. 59, 341 (1981)
11. Koukaras, E.N., Zdetsis, A.D., Karamanis, P., Pouchan, C., Avramopoulos, A., Papado Poulos, M.G.: J. Comp. Chem. 33, 1068 (2012)
12. Papadopoulos, M.G., Raptis, S.G., Demetropoulos, I.N.: Mol. Phys. 92, 547 (1997)
13. Ingamells, V.E., Raptis, S.G., Avramopoulos, A., Papadopoulos, M.G.: The Unusual effect of lithium substitution in organic molecules: The polarizability and second hyperpolarizability of $C_2H_2Li_2$. In: Papadopoulos, M.G. (ed.) Nonlinear Optical Responses of Molecules, Solids and Liquids: Methods and Applications, Research Singpost, pp. 97–111 (2003)

14. Papadopoulos, M.G., Raptis, S.G., Demetropoulos, I.N., Nasiou, S.M.: Theor. Chim. Acc. 99, 124 (1998)
15. Theologitis, M., Screttas, G.C., Raptis, S.G., Papadopoulos, M.G.: Int. J. Quant. Chem. 72, 177 (1999)
16. (a) Stewart, J.J.P.: J. Comp. Chem. 10, 209 (1989); (b) Anders, E., Koch, R., Freunscht, P.: J. Comp. Chem. 14, 1301 (1993)
17. Raptis, S.G., Papadopoulos, M.G., Sadlej, A.J.: Phys. Chem. Chem. Phys. 2, 3393 (2000)
18. Shimp, L.A., Chung, C., Lagow, R.J.: Inorg. Chim. Acta 29, 77 (1978)
19. Baran Jr., R.J., Hendrickson, D.A., Laude Jr., D.A., Lagow, R.J.: J. Org. Chem. 57, 3759 (1992)
20. Schmidt, M.W., Baldridge, K.K., Boatz, J.A., Elbert, S.T., Gordon, M.S., Jensen, J.H., Koseki, S., Matsunaga, N., Nguyen, K.A., Su, S.J., Windus, T.L., Dupuis, M., Montgomery, J.A.: J. Comput. Chem. 14, 1347 (1993); Granovsky, A.A.: The particular version of GAMESS used in our calculations is the PC GAMESS version 4.5, build number 1380. Partial Copyright 1994, Moscow State University, Moscow (1998)
21. Avramopoulos, A., Serrano-Andrés, L., Li, J., Papadopoulos, M.G.: J. Chem. Theor. Comp. 6, 3365 (2010)
22. Avramopoulos, A., Serrano-Andrés, L., Li, J., Reis, H., Papadopoulos, M.G.: J. Chem. Phys. 127, 214102 (2007)
23. Berski, S., Sivi, B., Lundell, J., Noury, S., Latajka, Z.: New Trends in Quantum Systems. In: Maruani, J., Minot, C., McWeeny, R., Smeyers, Y.G., Wilson, S. (eds.) Chemistry and Physics, vol. 1, p. 259. Kluwer Academic Publishers, London (1999)
24. Brown, E., Cohen, A., Gerber, R.B.J.: Chem. Phys. 122, 171101 (2005)
25. Khriachtchev, L., Taskanen, H., Lundell, J., Pettersson, M., Kiljunen, H., Räsänen, M.: J. Am. Chem. Soc. 125, 4696 (2003)
26. Papadopoulos, M.G., Avramopoulos, A.: Computational Methods in Science and Engineering. In: Maroulis, G., Simos, G. (eds.) Theory and Computation: Old Problems and New Challenges, CP963, 316 p. American Institute of Physics (2007)
27. Pettersson, M., Lundell, J., Räsänen, M.: Eur. J. Inorg. Chem. 729 (1999)
28. Khriachtchev, L., Pettersson, M., Runeberg, N., Lundel, J., Räsänen, M.: Nature 406, 874 (2000)
29. Wong, M.W.: J. Am. Chem. Soc. 122, 6289 (2000)
30. Avramopoulos, A., Reis, H., Li, J., Papadopoulos, M.G.: J. Am. Chem Soc. 126, 6179 (2004)
31. Oudar, L.J., Chemla, D.S.: J. Chem. Phys. 66, 2664 (1977)
32. Li, J., McWeeny, R.: Int. J. Quantum Chem. 89, 208 (2002)
33. Holka, F., Avramopoulos, A., Loboda, O., KellÖ, V., Papadopoulos, M.G.: Chem. Phys. Lett. 472, 185 (2009)
34. Cooke, S.A., Gerry, M.C.L.: Phys. Chem. Chem. Phys. 6, 3248 (2004)
35. Thomas, J.M., Walker, N.R., Cooke, S.A., Gerry, M.C.L.: J. Am. Chem. Soc. 126, 1235 (2004)
36. Walker, N.R., Reynard, L.M., Gerry, M.C.L.: J. Mol. Struct. 612, 109 (2002)
37. Michaud, J.M., Cooke, S.A., Gerry, M.C.L.: Inorg. Chem. 43, 3871 (2004)
38. Lundell, J., Ghaban, G.M., Gerber, R.B.: Chem. Phys. Lett. 331, 308 (2000)
39. Cooke, S.A., Gerry, M.C.L.: J. Am. Chem. Soc. 126, 17000 (2004)
40. Ghanty, T.K.: J. Chem. Phys. 123, 074323 (2005)
41. Pettersson, M., Lundell, J., Räsänen, M.: J. Chem. Phys. 102, 6423 (1995)
42. Peterson, K.A., Puzzarini, C.: Theor. Chim. Acc. 114, 283 (2005)

43. Peterson, K.A., Figgen, D., Goll, E., Stoll, H., Dolg, M.: J. Chem. Phys. 119, 11113 (2003)
44. Stein, L., Norris, J.R., Downs, A.J., Minihan, A.R.: J. Chem. Soc. Chem. Commun. 502, 502 (1978)
45. Drews, T., Seppelt, K.: Angew: Chem. Int. Ed. 36, 273 (1997)
46. Jiménez-Halla, C.O.C., Fernández, I., Frenking, G.: Angew: Chem. Int. Ed. 48, 366 (2009)
47. Cadet, P.E., Huo, W.M.: J. Chem. Phys. 47, 614 (1967)

Some Aspects of the Comparative Study of Semi-empirical Combustion Models on FLUENT and OpenFOAM Codes

Sreten Lekić[1], Mihajlo Savić[2], and Petar Gvero[3]

[1] Faculty of Natural Sciences and Mathematics,
University of Banja Luka, Bosnia and Herzegovina
[2] Faculty of Electrical Engineering,
University of Banja Luka, Bosnia and Herzegovina
[3] Faculty of Mechanical Engineering,
University of Banja Luka, Bosnia and Herzegovina
badaboom@etfbl.net

Abstract. Development of an optimal medium power gas burner was carried out under the EU FP6 project. Modeling and simulation was performed using the FLUENT and partly StarCD Adapco software packages. Restrictions that were imposed by available resources and 32-bit architecture allowed us to simulate only a small segment on the assumption of complete radial symmetry. In HP-SEE project we used available supercomputers and simulated the full cross section of the burner in OpenFOAM. An analysis of the differences in the results was performed. The advantages of this approach to CFD modeling of combustion were demonstrated.

Keywords: fluid dynamics, combustion, parallel processing.

1 Introduction

Use of high-performance computing resources in HP-SEE project gave us the opportunity to simulate the complete burner and to design and build more realistic model of real device. Simulations were performed by OpenFOAM[1] software version 1.7.1 with combustion and chemistry modules. The size of the mesh is over 13 million elements. Parallel processing of the partitioned model requires approximately 1 GB of RAM per core involved in computation at full scale. Special problems are presented in dividing the domain in such a way to enable all the CPUs to be optimally utilized. Simulation results verify the designed prototype burner. Test results will serve to optimize the parameters of an improved burner.

Simulation of gaseous combustion processes in cylindrical burner was made on full geometry shape (not only on one segment as before, on 32 bit platforms running small scale simulations). Good stability was achieved and additional mechanisms were included in simulation. Simulation was tested on up to 64 processors with almost linear improvement in simulation performance but limited

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_11, © Springer International Publishing Switzerland 2014

speed-up in pre- and post-processing. Suggestion is to perform these operations off-line prior to starting the simulation on supercomputing resources. Simulations running on heterogeneous cluster over MPI on four nodes performed as expected. Performance issues are foreseen on higher number of nodes with simple Gigabit Ethernet networking. Simulations were executed successfully with results aligning to expected values. We did encounter some issues: when dealing with large number of processors special care needs to be taken in mesh partitioning in order to avoid exceptions and crashes when running simulation; careful choice of various parameters is critical as simulation can easily produce several hundreds of gigabytes of files; networking performance can cause issues when scaling to higher number of nodes on Ethernet based networks.

2 Combustion Mechanisms

In OpenFOAM packages, stationary modules use the mechanisms of the mixing and transport of chemical species by solving conservation equations describing convection, diffusion, and reaction sources for each component species[2]. Multiple simultaneous chemical reactions can be modeled, with reactions occurring in the bulk phase (volumetric reactions) and/or on wall or particle surfaces.

2.1 Theory – Species Transport Equations

When we choose to solve conservation equations for chemical species [5,3], program predicts the local mass fraction of each species [1], Y_i, through the solution of a convection-diffusion equation for the ith species. This conservation equation takes the following general form:

$$\frac{\partial}{\partial t}\rho(Y_i) + \nabla \cdot (\rho \boldsymbol{v} Y_i) = -\nabla \cdot \boldsymbol{J}_i + R_i + S_i \tag{1}$$

where R_i is the net rate of production by chemical reaction and S_i is the rate of creation by addition from the dispersed phase plus any defined sources. An equation of this form will be solved for N - 1 species where N is the total number of fluid phase chemical species present in the system [4,5]. Since the mass fraction of the species must sum to unity, the Nth mass fraction is determined as one minus the sum of the N - 1 solved mass fractions. To minimize numerical error, the Nth species should be selected as that species with the overall largest mass fraction, such as N_2 when the oxidizer is air.

Mass Diffusion in Laminar Flows
In Equation (1), J_i is the diffusion flux of ith species, which arises due to concentration gradients. By default, program uses the dilute approximation, under which the diffusion flux can be written as

$$\boldsymbol{J}_i = -\rho D_{i,m} \nabla Y_i \tag{2}$$

J_i is the diffusion flux of ith species, which arises Here $D_{i,m}$ is the diffusion coefficient for ith species in the mixture. For certain laminar flows, the dilute

approximation may not be acceptable, and full multicomponent diffusion is required. In such cases, the Maxwell-Stefan equations can be solved.

Mass Diffusion in Turbulent Flows
In turbulent flows [4,5],s program computes the mass diffusion in the following form:

$$\boldsymbol{J}_i = -\left(\rho D_{i,m} + \frac{\mu_t}{Sc_t}\right)\nabla Y_i \tag{3}$$

where Sc_t is the turbulent Schmidt number, and $\frac{\mu}{\rho D_t}$ has a default setting 0.7. Turbulent diffusion generally overwhelms laminar diffusion, and the specification of detailed laminar diffusion properties in turbulent flows is not warranted.

The Eddy-Dissipation-Concept (EDC) model is an extension of the eddy dissipation model to include detailed chemical mechanisms in turbulent flows. It assumes that reaction occurs in small turbulent structures, called the fine scales. Detailed chemical kinetic calculations can be very computationally expensive.

3 Combustion Simulation Results

The redesigned model has included the effects of radiation using the DO mechanism [5,6]. Chemical processes are supported in Species Combustion module in CHEMKIN data format. Currently, the module for NOx production is included. Implementation of all modules is used for more detailed study of combustion kinetics. Development of reduced combustion kinetic mechanism and its application to optimize the emission of pollutants is goal scientific result [7,6]. There was made further development and optimization of the burner prototype.

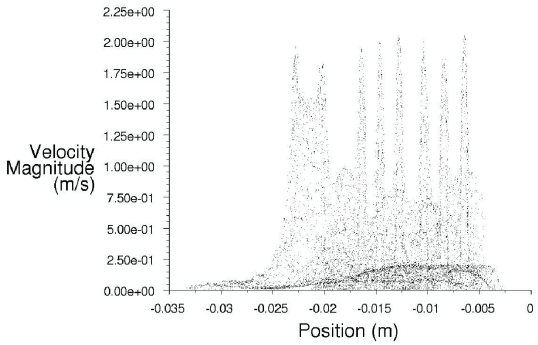


Fig. 1. Distribution of velocity magnitude along the slices

4 Scalability Testing

For scalability testing we have used OpenFOAM[1] version 1.7.1 at PARADOX cluster located at Institute of Physics Belgrade. The cluster is composed of 84

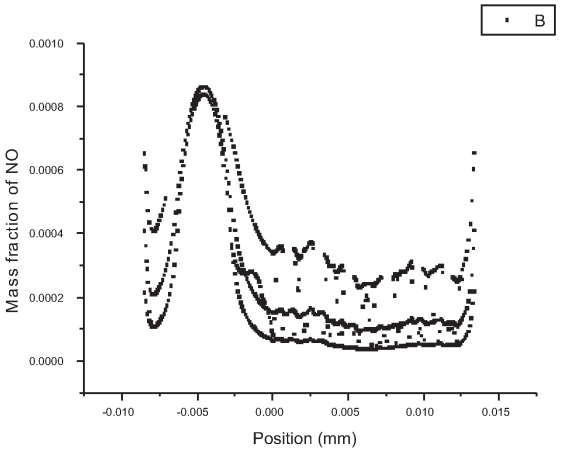


Fig. 2. Distribution of NOx mol concentration along the slices

nodes with each node consisting of two quad core Intel Xeon E5345 processors running at 2.33 GHz with 8GB of RAM. Each node is connected by dual Gigabit Ethernet connection to form star topology network. OpenFOAM was compiled by GNU Compiler Collection 4.4.6 and OpenMPI 1.4.1 was used for MPI.

We used simplified smaller scale methane burner and *rhoReactingFoam* solver for benchmarking. Total model size was 811 MB and mesh contained 1893571 points, 18913870 faces and 9132732 tetrahedral cells.

4.1 Execution Times

As can be seen from the table with execution times, there are two dominant factors: pre/post-processing and simulation. Pre-processing and especially post-processing time increases with number of CPU cores used for simulation but, fortunately, the increase is not dramatic. It is worth noting that pre- and post-processing can be avoided in certain work-flows allowing for better scalability.

Table 1. Scalability testing of SFHG application

CPU cores	Pre/Post[s]	Simulation[s]	Total[s]	Simulation Speedup	Total Speedup
1	0.00	127137.60	127137.60	1.00	1.00
4	2560.00	33922.80	36482.80	3.75	3.48
8	2650.72	17496.00	20146.72	7.27	6.31
16	2817.06	8359.20	11176.26	15.21	11.38
32	3140.02	4135.05	7275.07	30.75	17.48

Fig. 3. Burner mantle CAD model

4.2 Scalability Analysis

Since this application was based on tried and true solvers that have already been thoroughly benchmarked and analyzed, we focused our attention to issues specific to this concrete case.

I/O operations were affecting mostly pre- and post-processing parts of the work-flow, but not in significant amount. Communication between processes can be minimized by using adequate mesh partitioning approach. For generic use case we would suggest using scotch algorithm which is designed to minimize boundaries shared between different partitions. If there is sufficient knowledge of concrete simulation and model behavior better results can be achieved by using hierarchical or, better yet, manual decomposition.

When discussing scalability of simulation, one has to take into consideration several factors. If we are dealing with cold flow simulations, with no chemistry involved, then it is fairly easy to achieve good scalability of simulation by adequate mesh partitioning. But, when we include chemistry in simulation things start to get complicated as chemistry is generally more time consuming to simulate, especially when dealing with complex reactions and more realistic models. During the times close to the beginning of the simulation, computationally intensive part of the simulated model is fairly small and located at inlet and as such using huge number of CPUs would bring severely limited benefits. If increasing performance is a must even a this stage, one must take care of properly partitioning the mesh so that we do not end up with many processes waiting for few with more complex calculations to handle. After a while, or 1.3 seconds of simulated time in our case, most of the model volume is involved in time

consuming chemistry calculations and as such it becomes easier to decompose the mesh in such a way to achieve good scalability.

Above analysis applies foremost to solvers that work in time domain. When we are dealing with steady-state solvers, according to our results, they tend to reach an analogous point in fairly short amount of time and are as such better, or at least easier, choices for good scalability.

5 Summary and Conclusion

A good agreement in calculated parameters of combustion processes between OpenFOAM and Fluent was achieved after careful parameter tuning. OpenFOAM demonstrated good capabilities when compared to commercial software such as Fluent and StarCD Adapco. Parallelization allows for very good performance and scalability of OpenFOAM package and enables avoiding the high costs of the end user licenses of commercial software.

Acknowledgement. This work makes use of results produced by the High-Performance Computing Infrastructure for South East Europes Research Communities (HP-SEE), a project co-funded by the European Commission (under contract number 261499) through the Seventh Framework Programme. HP-SEE involves and addresses specific needs of a number of new multi-disciplinary international scientific communities (computational physics, computational chemistry, life sciences, etc.) and thus stimulates the use and expansion of the emerging new regional HPC infrastructure and its services.

References

1. OpenFOAM, `http://www.openfoam.org/`
2. Baulch, D.L., Drysdall, D.D., Horne, D.G., Lloyd, A.C.: Evaluated Kinetic Data for High Temperature Reactions, vol. 1,2,3. Butterworth (1973)
3. Baxter, L.L.: Turbulent Transport of Particles. PhD thesis, Brigham Young University, Provo, Utah (1989)
4. Binniger, B., Chan, M., Paczkko, G., Herrmann, M.: Numerical Simulation of Turbulent Partially Premixed Hydrogen Flames with the Flamelet Model. Technical report, Advanced Combustion Gmbh, Internal Report (1998)
5. Launder, B.E., Spalding, D.B.: The Numerical Computation of Turbulent Flows. Computer Methods in Applied Mechanics and Engineering 3, 269–289 (1974)
6. Barnes, F.J., Bromly, J.H., Edwards, T.J., Madngezewsky, R.: NOx Emissions from Radiant Gas Burners. Journal of the Institute of Energy 155, 184–188 (1988)
7. Ahmad, T., Plee, S.L., Myers, J.P.: Computation of Nitric Oxide and Soot Emissions from Turbulent Dffusion Flames. J. of Engineering for Gas Turbines and Power 107, 4853 (1985)

Development of a Hybrid Statistical Physics – Quantum Mechanical Methodology for Computer Simulations of Condensed Phases and Its Implementation on High-Performance Computing Systems*

Ljupčo Pejov[1], Dragan Sahpaski[2], Emilija Kohls[1], and Anastas Mishev[2]

[1] University Sts Cyril and Methodius, Faculty of Natural Sciences & Mathematics Institute of Chemistry, Skopje, Macedonia
[2] University Sts Cyril and Methodius, Faculty of Computer Sciences & Engineering, Skopje, Macedonia
ljupcop@pmf.ukim.mk, anastas.mishev@finki.ukim.mk

Abstract. A hybrid, complex statistical mechanics – quantum mechanical approach which enables exact computational modeling of condensed phases at finite temperatures has been developed and implemented on high-performance computing systems. The computational approach is robust and inherently sequential. First, the studied physico-chemical system is modeled by a statistical physics approach, either Monte Carlo (MC) or molecular dynamics (MD). Though in the first phase it is often sufficient to carry out a classical MC or MD simulation, in particular cases, when it is necessary, one can also perform a quantum molecular dynamics simulation (e.g. ADMP, BOMD or CPMD). Even the classical MC/MD simulations carried out in the first phase can be based on interaction potentials which have been derived by quantum chemical calculations. Sequentially to the first phase of the computation, which actually generates either a MD trajectory or an appropriate sample of the system's configurational space, the generated trajectories are analyzed employing time-series analytic methods. On the basis of such analysis, a representative number of configurations representing the state of the physico-chemical system at finite temperature is chosen, which are further analyzed by a quantum mechanical approach. These, appropriately chosen configurations for the system of interest, are further modeled by exact quantum mechanical (QM) approach. The particular approach that needs to be implemented depends on the quantity that needs to be computed. For example, in the case of X-H stretching vibrations, the anharmonic X-H vibrational frequency is computed in a quantum mechanical manner with respect to both electronic and nuclear subsystem. In this case, first a 1D cut through the vibrational potential energy surface is computed at series of suitably generated points, and subsequently, the vibrational Schrodinger equation is solved either by diagonalization approach or using a variant of the discrete variable representation (DVR) methodology. As the statistical mechanics simulations are often done implementing periodic

* This paper is based on the work done in the framework of the HP-SEE FP7 EC funded project.

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_12, © Springer International Publishing Switzerland 2014

boundary condition, the exact QM calculations in the final simulation phase are often done implementing some sort of embedding of the relevant part of the system. All these aspects of our developed methodology are illustrated through a particular example – the fluoroform solvated in liquid Kr and the noncovalently bonded complexes which are formed between fluoroform and dimethylether in liquid Kr. A vast variety of condensed phase systems can be treated by the developed approach. Achieving good parallel efficiency for calculations of such type is far from a trivial task without the use of high-performance low-latency MPI interconnect (such as, e.g. a supercomputer or HPC cluster).

Keywords: molecular dynamics, Monte Carlo, condensed phases, liquids, fluctuating environments, Monte Carlo, molecular dynamics, intermolecular interaction potentials, hybrid statistical physics – quantum mechanical approach.

1 Introduction

One of the most widely used approaches in molecular quantum mechanics has been based on exploration of potential energy surfaces of individual molecules and molecular aggregates/clusters. Even nowadays, numerous quantum theoretical studies of condensed-phase systems aim to simulate the many-particle phases by exploring the PES of a particular cluster, mimicking the condensed phase in question. This approach, though being straightforwardly and easily applicable throughout the usage of standard quantum mechanics computational packages, has certain drawbacks and inherent limitations. First of all, when one aims to describe a complex condensed-phase system by a finite-cluster approach, certain dynamical properties are neglected at the very beginning. Having in mind that condensed-phase systems are of substantial importance in both fundamental natural sciences and technology, appropriate theoretical modeling of such systems has been shown to be of crucial importance for a thorough understanding of their properties. Numerous examples exist in the literature in which it has been even demonstrated that a particular theoretical model can provide complementary information to those obtained by the experimental studies [1]. Especially for technological applications of condensed-phase systems, it is crucial to understand their properties at finite temperatures, while virtually all studies based on PES explorations refer to 0 K. By such treatment, it is not possible to explicitly include the molecular motions in the (correct) description of the system of interest. In the present paper, we aim to implement a general hybrid methodology which explicitly accounts for the dynamical phenomena in the description of the condensed phases. As a particular example, we focus our attention on solvation of fluoroform and fluoroform – dimethylether in liquid Kr at 131 K. As a property particularly sensitive to fluctuating in-liquid environment, we pay a special attention to anharmonic C-H stretching vibrational frequency shift of the fluoroform moiety upon solvation and complexation. We also actually demonstrate that X-H stretching frequencies are a particularly good indicator whether the intermolecular interaction potentials are good enough to predict accurately the structure of liquid in the immediate vicinity of the solute molecule.

2 Computational Details and Algorithms

The statistical physics methodology implemented for the particular purpose of the present study was Monte Carlo (MC) approach [2]. In all MC simulations in the present study, intermolecular interactions were described by a sum of Lennard-Jones 12-6 site-site interaction energies plus Coulomb terms:

$$U_{ab} = \sum_{i}^{a}\sum_{j}^{b} 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \frac{q_i q_j e^2}{4\pi\varepsilon_0 r_{ij}} \tag{1}$$

where i and j are sites in interacting molecular systems a and b, r_{ij} is the interatomic distance between sites i and j, while e is the elementary charge. The following combination rules were used to generate two-site Lennard-Jones parameters ε_{ij} and σ_{ij} from the single-site ones:

$$\varepsilon_{ij} = \sqrt{\varepsilon_i \varepsilon_j} \tag{2}$$

$$\sigma_{ij} = \sqrt{\sigma_i \sigma_j} \tag{3}$$

Model potential parameters σ = 3.895 Å and ε = 0.308 kcal mol^{-1} were used For Kr, while the charge distribution in the case of fluoroform and dimethylether (DME) was computed in the following manner. The calculated MP2/6-31++G(*d*,*p*) electronic density (corresponding to the minima on the MP2/6-31++G(*d*,*p*) PESs) was fitted to a set of point charges placed at the nuclear positions using the CHelpG point-selection algorithm [3]. The fitting procedure was carried out imposing a constrain that the point-charge distribution reproduces the "correct" molecular dipole moment computed from the MP2/6-31++G(*d*,*p*) electronic density for each molecule. Geometries corresponding to the minima on MP2/6-31++G(*d*,*p*) PESs of fluoroform and dimethylether were used throughout the rigid-body Monte Carlo simulations. Non-bonded LJ parameters for these two molecules were taken from the OPLS-AA force field [4]. The long-range corrections (LRC) to the interaction energy were calculated for interacting atomic pairs between which the distance is larger than the cutoff radius defined as half of the unit cell length. The Lennard-Jones contribution to the interaction energy beyond this distance was estimated assuming uniform density distribution in the liquid (*i.e.* $g(r) \approx 1$), while the electrostatic contribution was estimated by the reaction field method involving the dipolar interactions. MC simulations were performed in the canonical (*NVT*) ensemble, implementing the Metropolis sampling algorithm, at T = 131 K, using the experimental density of liquid Kr of 2.3406 g cm^{-3} at these conditions. These particular conditions actually correspond to the experimental ones under which the Raman spectra of CF_3H and CF_3H – dimethylether in liquid Kr cryosolutions were recorded [5]. We have carried out a MC simulation of one fluoroform molecule plus 1249 Kr atoms placed in a cubic box with side length of 42.04 Å, and also of one fluoroform and one dimethylether molecules plus 1249 Kr atoms placed in a cubic box with a = 42.05 Å, imposing periodic boundary conditions. All simulations consisted of thermalization phase of at least $6.25 \cdot 10^7$ MC steps, subsequently followed by averaging (simulation)

phase of at least $2.50 \cdot 10^8$ MC steps. MC simulations were performed with the DICE statistical mechanics Monte Carlo suite of codes [6].

To compute the anharmonic C-H Stretching frequencies of fluoroform and fluoroform-dimethylether in liquid Kr, series of configurations (snapshots) from the equilibrated MC runs were used to calculate the "in-liquid" C-H stretching potential of the fluoroform molecule. This was done by a quantum-mechanical (QM) approach, or, more precisely, $QM_{electronic}$ + QM_{nuclei}, as both the electronic and nuclear subsystems were treated quantum mechanically [7]. As the computational cost of the sequential QM phase is relatively high, it is advisable to use a relatively small number of structures generated by the statistical physics simulations. These structures, however, should represent appropriately the configurational space sampled by the Metropolis technique. Having in mind that the statistical correlation between MC-generated configurations which are sufficiently close to each other is comparatively high, performing QM calculations on such configurations would be a waste of time, as they won't add any new statistically relevant contribution to the results. As explained in a series of works of Coutinho *et. al.* [8], a much better approach is to choose configurations with low mutual statistical correlation and perform the QM computational part only on these configurations. The choice of statistically uncorrelated configurations is essentially based on time series analysis methods. In the case of Markovian random process (such as the MC chain generated by the Metropolis sampling algorithm), the energy autocorrelation function may serve as a clear indicator for mutual statistical correlation between subsequent MC-generated configurations [8].We have therefore computed $C(n)$ from the MC simulation phase of CF_3H and $CF_3H\ldots(CH_3)_2O$ and 1249 Kr atoms, together with the exponential decay function fits (the plots are available from the authors upon request). By integration of the energy autocorrelation functions, correlation steps of about 111 and 128 for the two cases were obtained, respectively. In both cases, MC configurations mutually separated by 500 steps ($> 4 \cdot \tau$) are correlated less than 10 %. For sequential QM calculations which were further carried out in the present study from this periodic box, we chose 50 uncorrelated configurations from the equilibrated MC runs, separated by as much as 3500 MC steps. In that case, the mutual correlation between the configurations is negligible (less than 0.01 %). The $QM_{electronic}$ + QM_{nuclei} calculations were carried out for 50 supermolecular clusters containing one fluoroform molecule (or a single fluoroform…dimethylether dimer) and a representative part of the first solvation shell around fluoroform (or the fluoroform…dimethylether dimer). A give Kr atom was included in the relevant part of the first solvation shell if the distance between the Kr atom and the fluoroform H atom was smaller than 4.0 Å.

Vibrational potential energy function ($V = f\ (r_{OH})$) for each CH oscillator corresponding to a particular representative MC configuration was computed by carrying out a series of 20 pointwise HF, DFT or MP2 energy calculations. Throughout these calculations, the C-H distances were varied from 0.900 to 1.375 Å with a step of 0.025 Å, keeping the center-of-mass of the vibrating C-H fragment fixed, in order to mimic as closely as possible the real normal mode. The obtained energies were least-squares fitted to a fifth-order polynomial in Δr_{OH} ($\Delta r = r - r_e$):

$$V = V_0 + k_2 \Delta r^2 + k_3 \Delta r^3 + k_4 \Delta r^4 + k_5 \Delta r^5 \quad (4)$$

The resulting potential energy functions were subsequently cut after fourth order and transformed into Simons-Parr-Finlan (SPF) type coordinates [9]:

$$\rho = 1 - r_{OH,e}/r_{OH} \quad (5)$$

(where $r_{OH,e}$ is the equilibrium, *i.e.* the lowest-energy, value). The one-dimensional vibrational Schrödinger equation was solved variationally and the fundamental *anharmonic* C-H stretching frequency was computed from the energy difference between the ground and first excited vibrational states.

In the case of in-liquid C-H stretching frequency values, the *density-of-states (DOS) histograms*, which are a good approximation to the experimentally observed Raman bands were subsequently generated from the calculated *anharmonic* C-H stretching frequencies for the MC-generated in-solutions configurations. All quantum-chemical (electronic) calculations in the present study were performed with the Gaussian03 series of codes [10].

3 Results and Discussion

For a free fluoroform molecule, applying the described method of computation, we obtained a value of 3227.5 cm^{-1} for the fundamental anharmonic C-H stretching frequency at the HF/6-31++G(*d*,*p*) level of theory, 3002.9 cm^{-1} at B3LYP/6-31++G(*d*,*p*) and 3110.7 cm^{-1} at MP2/6-31++G(*d*,*p*) levels (the values referring to the corresponding minima on the PESs). The corresponding experimental value is 3035.2 cm^{-1} [11].

The C-Kr radial distribution function (RDF) for CF_3H in liquid Kr, computed from the equilibrated Monte Carlo run is shown in Fig. 1.

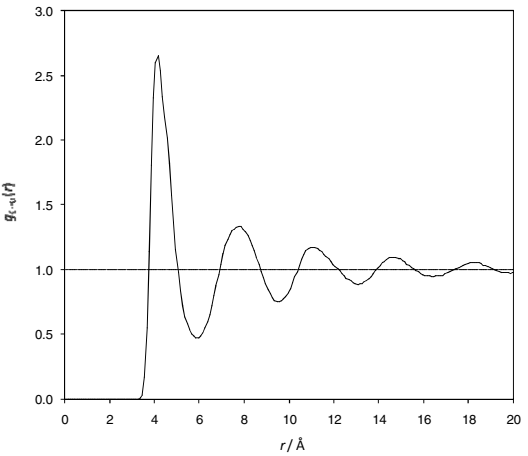


Fig. 1. The radial distribution function between the fluoroform C atom and Kr (solvent) computed from MC simulations

As much as five solvation shells are clearly visible under the particular conditions. The first solvation shell starts at about 3.5 Å and ends at about 6 Å. It contains 14 Kr atoms on average. In Fig. 2, several configurations extracted from the MC run are shown, with the central fluoroform molecule and several Kr atoms, satisfying the condition mentioned above.

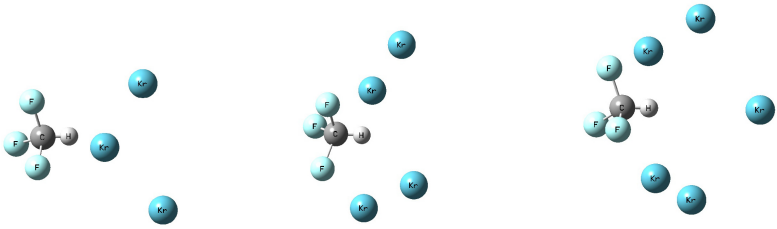


Fig. 2. Several configurations extracted from the MC simulation run for fluoroform in liquid Kr at 131 K

The density of states histogram generated from the computed in-liquid C-H stretching frequencies of fluoroform in liquid Kr by the B3LYP method is shown in Fig. 3. As can be seen, the advanced computational method applied in the present study predicts a blue shift of the fluoroform CH stretching frequency (Table 1) upon its solvation in liquid Kr at 131 K. Such result is in contrast with the experimental data, according to which the CH stretching frequency experiences a small red shift (~ – 2 cm^{-1}) upon dissolving CF_3H in liquid Kr.

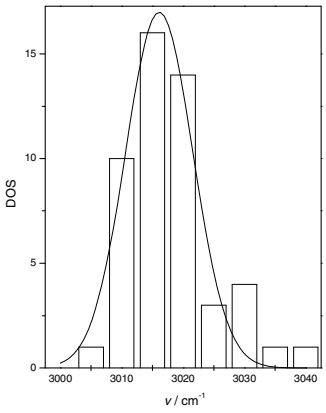


Fig. 3. The density of states (DOS) histogram generated from the computed in-liquid C-H stretching frequencies of fluoroform in liquid Kr by the B3LYP method

The question is: can one explain such apparent discrepancy in a plausible way? To achieve this goal, we adopt the following approach. The in-solution X-H stretching frequencies are highly sensitive to the local “in-liquid” environment of the oscillator, but also depend on the long-range electrostatic effects (the “bulk” part of the solvent).

In the present case of a non-polar solvent, only the nearest-neighbor effects are relevant. According to our test calculations, the C-H stretching frequency of solute is most sensitive to the presence of Kr atoms on the H-atom side. Thus, the overall CH stretching frequency shift would be primarily determined by the distribution of Kr atoms in the first solvation shell around CF_3H, in particular their distance from the vibrating hydrogen.

Table 1. Anharmonic C-H stretching frequencies of free CF_3H, CF_3H solvated in liquid Kr and CF_3H-DME dimer solvated in liquid Kr. The corresponding frequency shifts with respect to the free molecule are given in parentheses. All calculations were done with the 6-31++G(*d*,*p*) basis set (see text for details).

Level of theory	Free CF_3H	CF_3H in liquid Kr	CF_3H…DME in liquid Kr
	ν / cm^{-1}	$\langle\nu\rangle$ / cm^{-1}	$\langle\nu\rangle$ / cm^{-1}
MP2	3110.7	3120.2 (+9.5)	3131.0 (+20.3)
B3LYP	3003.5	3017.6 (+14.1)	3031.5 (+28.0)
HF	3234.6	3257.0 (+22.4)	3274.4 (+39.8)
Exp.	3035.2	3033.1 (–2.1)	3051.8 (+16.6)

To clarify further this point, we have computed the anharmonic C-H stretching frequency as a function of the C…Kr distance (R) for the CF_3H…Kr dimer upon a collinear approach of Kr atom with the C-H bond. The results are schematically presented in Fig. 4.

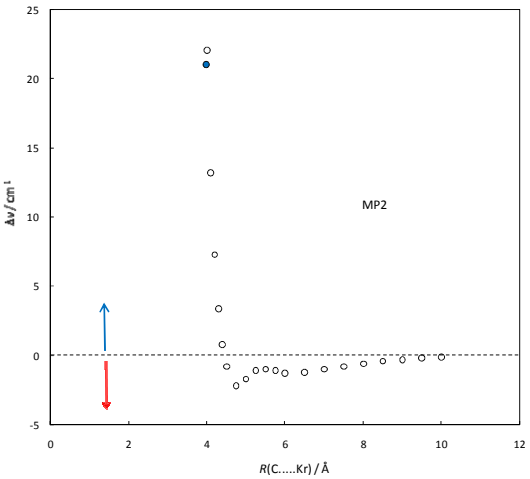


Fig. 4. The anharmonic C-H stretching frequency as a function of the C…Kr distance (R) for the CF_3H…Kr dimer upon a collinear approach of Kr atom with the C-H bond, computed at MP2 level of theory

As seen from Fig. 4, MP2 level of theory predicts a small red shift of the CH stretching frequency as the Kr atom approaches CF_3H along the C-H axis. This shift, upon certain fluctuations, even becomes larger (in absolute value) as the distance *R* decreases, reaches a minimum, and then starts to rise abruptly at distances *R* smaller than about 4.75 Å. Having in mind that the maximum of the C…Kr radial distribution function computed from MC simulation falls at about 4.15 Å, it is obvious that most of the Kr atoms within the first shell are located at positions at which the CH frequency shifts acquire either small negative or small positive values according to Fig. 4. It so happens, therefore, that even the actual *sign* of the frequency shift (*i.e.* its very classification as being a red or blue shift) is strongly dependent on the actual value of *R*, *i.e.* the correct description of the first solvation shell around CF_3H by the Monte Carlo method. The description of the structure of the liquid, on the other hand, is strongly dependent on the quality of the intermolecular interaction potentials. The currently used potentials could be considered as being accurate enough for certain other purposes, but it appears that particularly correct description of the distribution of solvent molecules around solute is crucial for a correct prediction of the vibrational frequency shifts. *Anharmonic* vibrational frequencies computed by actual scanning of the potential energy surface cut along a particular vibrational coordinate necessary involve movement of atoms within the solute molecule. Particularly in the case of higher amplitude motions, such as those involving light atoms, the oscillator probes (*i.e.* in a sense feels) the environment more closely than it is the case in other molecular properties.

The C_{CF3H} - O_{DME} radial distribution function, computed from the MC simulation of the CF_3H-DME dimer in liquid Kr is shown in Fig. 5.

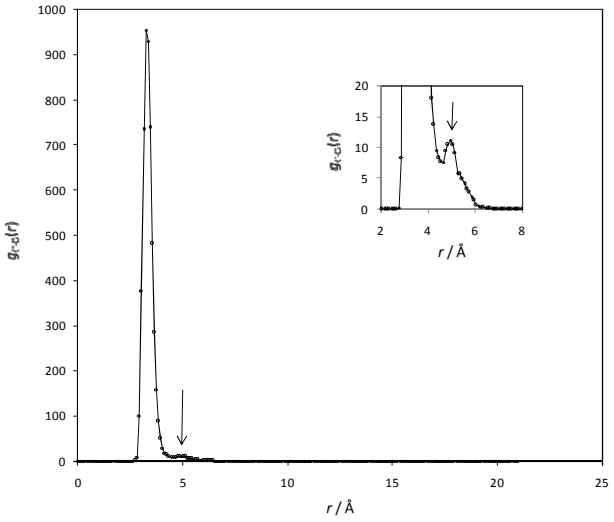
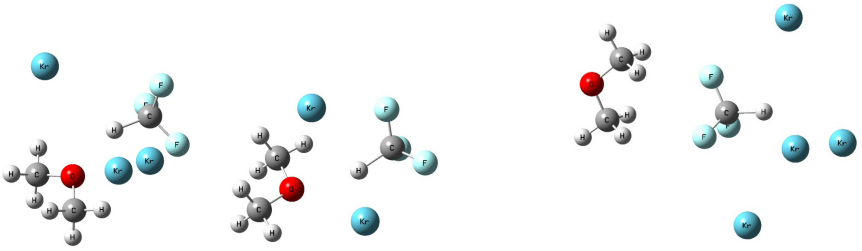


Fig. 5. The C_{CF3H} - O_{DME} radial distribution function computed from MC simulations

Two populations of dimeric structures are distinguishable in liquid Kr, which exhibit reversible mutual interconversion. One of the two populations is obviously by far more predominant (corresponding to the higher maximum at about 3.25 Å), while the second one (corresponding to the maximum at about 4.95 Å) is much less important in the sense of its contribution to the spectroscopic properties. In Fig. 6, several configurations extracted from the MC run are shown, corresponding to both populations (a) and (b).



(a) (b)

Fig. 6. Several configurations extracted from the MC simulation run for fluoroform-DME dimer in liquid Kr at 131 K, corresponding to two populations (a) and (b)

The density of states histogram generated from the computed in-liquid C-H stretching frequencies of fluoroform-DME dimer in liquid Kr by the B3LYP method is shown in Fig. 7.

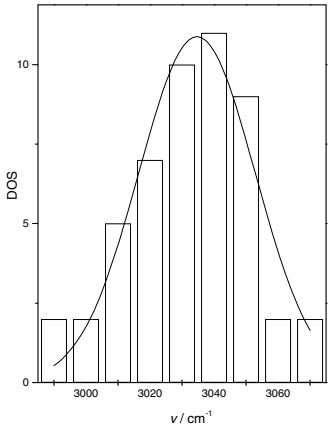


Fig. 7. The density of states (DOS) histogram generated from the computed in-liquid C-H stretching frequencies of fluoroform-DME dimer in liquid Kr by the B3LYP method

As can be seen from Table 1, MP2 level of theory performs best in the sense of quantitative prediction of the C-H stretching frequency shift upon complexation of fluoroform with dimethylether in liquid Kr (with respect to the gas-phase value).

4 Conclusions and Directions for Future work

In the present work, we have implemented a complex hybrid statistical physics - quantum mechanical methodology which is useful for simulation of complex condensed phase systems. The methodology is illustrated through a particular example – solvation of fluoroform and fluoroform – dimethylether in liquid Kr at 131 K. Particular attention was paid to exact calculations of anharmonic C-H stretching vibrational frequency shift of the fluoroform moiety upon solvation and complexation. It was found out that X-H stretching frequencies are a particularly good indicator whether the intermolecular interaction potentials are good enough to predict accurately the structure of liquid in the immediate vicinity of the solute molecule. The methodology is inherently a multistep one, and certain steps are excellently scalable on parallel high-performance clusters. Since it is difficult to fully automatize the overall procedure, judging the overall scalability of the algorithm is a complex task.

References

1. Pejov, L.: Chem. Phys. Lett. 376, 11 (2003)
2. Allen, M.P., Tildesley, D.J.: Computer Simulation of Liquids. Oxford University Press, New York (1997)
3. Breneman, C.M., Wiberg, K.B.: J. Comp. Chem. 11, 361 (1990)
4. Kaminski, G., Duffy, E.M., Matsui, T., Jorgensen, W.L.: J. Phys. Chem. 98, 13077 (1994)
5. Van den Kerkhof, T., Bouwen, A., Goovaerts, E., Herrebout, W.A., Van der Veken, B.J.: Phys. Chem. Chem. Phys. 6, 358 (2004)
6. Coutinho, K., Canuto, S.: DICE: a Monte Carlo Program for Molecular Liquid Simulation. University of São Paulo, São Paulo (2003)
7. Pejov, L., Spångberg, D., Hermansson, K.: J. Chem. Phys. 133, 174513 (2010)
8. Coutinho, K., Canuto, S.: Adv. Quantum Chem. 28, 89 (1997)
9. Simons, G., Parr, R.G., Finlan, J.M.: J. Chem. Phys. 59, 3229 (1973)
10. Frisch, M.J., Trucks, G.W., Schlegel, H.B., Scuseria, G.E., Robb, M.A., Cheeseman, J.R., Montgomery Jr., J.A., Vreven, T., Kudin, K.N., Burant, J.C., Millam, J.M., Iyengar, S.S., Tomasi, J., Barone, V., Mennucci, B., Cossi, M., Scalmani, G., Rega, N., Petersson, G.A., Nakatsuji, H., Hada, M., Ehara, M., Toyota, K., Fukuda, R., Hasegawa, J., Ishida, M., Nakajima, T., Honda, Y., Kitao, O., Nakai, H., Klene, M., Li, X., Knox, J.E., Hratchian, H.P., Cross, J.B., Adamo, C., Jaramillo, J., Gomperts, R., Stratmann, R.E., Yazyev, O., Austin, A.J., Cammi, R., Pomelli, C., Ochterski, J.W., Ayala, P.Y., Morokuma, K., Voth, G.A., Salvador, P., Dannenberg, J.J., Zakrzewski, V.G., Dapprich, S., Daniels, A.D., Strain, M.C., Farkas, O., Malick, D.K., Rabuck, A.D., Raghavachari, K., Foresman, J.B., Ortiz, J.V., Cui, Q., Baboul, A.G., Clifford, S., Cioslowski, J., Stefanov, B.B., Liu, G., Liashenko, A., Piskorz, P., Komaromi, I., Martin, R.L., Fox, D.J., Keith, T., Al-Laham, M.A., Peng, C.Y., Nanayakkara, A., Challacombe, M., Gill, P.M.W., Johnson, B., Chen, W., Wong, M.W., Gonzalez, C., Pople, J.A.: Gaussian 03 (Revision C.01). Gaussian, Inc., Pittsburgh (2003)
11. Dubal, H.R., Ha, T.K., Lewerenz, M., Quack, M.: J. Chem. Phys. 91, 6698 (1989)

Solvatochromic Effect for the Denaturation and Mutation Processes in DNA: Computational Study

Jumber Kereselidze[1,*], Marine Kvaraia[2], Zurab Pachulia[2], and George Mikuchadze[3]

[1] I. Javakhishvili Tbilisi State University,
3 I. Chavchavadze ave., 0179, Tbilisi, Georgia
jumber.kereselidze@tsu.ge
[2] Sokhumi State University,
9 Anna Politcovskaya str., 0186, Tbilisi, Georgia
[3] Georgian Research and Educational Networking Association (GRENA),
10 Chovelidze str, 0108, Tbilisi, Georgia
gmikuchadze@gmail.com

Abstract. The influence of the environment on the proton transfer between nucleotide bases has crucial importance for denaturation and mutation processes in DNA. For quantitative description of these processes, activation ($\Delta E^{\#}$) and reaction (ΔE) energies of the proton transfer as well as lactam-lactim (K_T^{LL}) and amino-imino (K_T^{AI}) tautomeric equilibrium constants by the quantum-chemical DFT method are calculated. It is shown that decrease in the environment polarity (E_r) due to mixing of ethanol with water (solvatochromic effect) leads to a decrease in the activation energy of the proton transfer and to an increase of the mutation frequency (v_m), and at the same time to the tendency of DNA to denaturation. Hence, energy and kinetic characteristics of the proton transfer may be used for quantitative estimation of a solvatochromic effect in DNA. The validity of the solvatochromic effect is confirmed by the bathochromic shift of the DNA absorption band in the UV spectrum.

Keywords: solvatochromic effect, DNA, denaturation, mutation, DFT calculations.

1 Introduction

It is well-known that water in DNA provides stabilization of canonical and rare tautomeric forms of nucleotide bases [1-4]. However, an decrease in the aqueous environment polarity due to mixing of ethanol, methanol and other organic solvents may cause DNA denaturation [5-7]. In particular, no double helix structure was found in a pure methanol solution [8]. Using the circular dichroism method, it is shown that in water-ethanol mixtures double helix destabilization increases with increasing ethanol content [9]. In quantitative description of the influence of the environment it is taken into consideration that the degree of hydration depends on the quantity of water molecules in the first DNA shell [10]. However, in this case, one should allow for the

* Corresponding author.

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_13, © Springer International Publishing Switzerland 2014

position of water molecules with respect to the functional groups of nucleotide bases involved in the proton transfer. Furthermore, it was found that hydration depends not only on the quantity, but also on the quality of solvent molecules [11]. The goal of the present paper is to build a quantum-chemical model of the solvatochromic effect and to study the mechanism of the influence of the environment on DNA mutation and denaturation processes.

2 Methods

For quantitative description of the influence of the environment polarity a quantum-chemical model of a complementary guanine-cytosin pair surrounded by six water molecules at the distance of 2 Å has been built. Subsequently, water molecules were gradually fully substituted by ethanol molecules(see Table 1). Depending on such a change in the environment, lactam-lactim KT (LL) and amino-imino KT (AI) tautomeric equilibrium constants, activation energy (ΔE#) and reaction energy (ΔE) of the proton transfer as well as point mutation frequency (νm) were calculated using the quantum-chemical density functional theory (DFT) method [12]. The calculations were performed by the computer program "Nature" [13] in the reaction coordinate regime (RNH) using the PBE approximation [14], its modification mPBE [15]and local density approximations [16] are promoted. The results of calculations, Reichardt values for the environment polarity parameter (Er) [17] and absorption bands in the UV spectrum in water (1) and ethanol (7) are presented in Table 1.

3 Implementation on the HPC Infrastructure

Before launching the application on the HPC clusters we ported it on the GRID infrastructure using our access on the SEE/EGEE-GRID. Only MPICH1 implementation of MPI was available for us on the GRID.

After successful porting on the GRID we launched the application on the NCIT (Politehnica University of Bucharest) HPC cluster in June 2011. We have been granted access to the HPC cluster with AMD Opteron 2435(6-core) cpus and Infiniband inter-node communication system (in total 168 cpu/core-s).

In Aug 2012 we launched the application on the Hungarian HPC clusters (Szeged Supercomputer Centre) with AMD Opteron 6174(12-core) processors and Infiniband inter-node communication system (summary more than 1000cpu/core-s).

We carried out scalability study of our application for various compilers and MPI implementations. It was obtained that the most optimized C/C++ compilers for the AMD Opteron processors are Open64 compiler v4.2.4 with AMD Core Math Library (ACML) at Szeged Supercomputer Centre and GCC v4.7.0 at NCIT. In both cases we have used openmpi v1.6 realization of MPI. The optimized number of cpu/core-s for the most of the investigated structures (number of atoms in the range 40-80) is 24. The benchmark results for the particular structure are given below.

```
NCIT-Cluster
working nodes: AMD Opteron 2435 (2.6Ghz)
mpi: openmpi v1.6
C compiler: GCC v4.7.0
# cores   Time (HH:MI:SS)    # nodes
16 cores           06:25:26        2
24 cores           05:33:34        2
32 cores           08:46:45        2

Szeged SC
working nodes: AMD Opteron 6174 (2.2Ghz)
mpi: openmpi v1.6
C compiler: Open64 Compiler Suite: v4.2.4 with AMD Core
Math Library (ACML)
# cores   Time (HH:MI:SS)    # nodes
08 cores          10:17:25            2
16 cores          07:54:30            6
24 cores          07:01:17            2
32 cores          08:45:08            7
64 cores          09:03:20           13
```

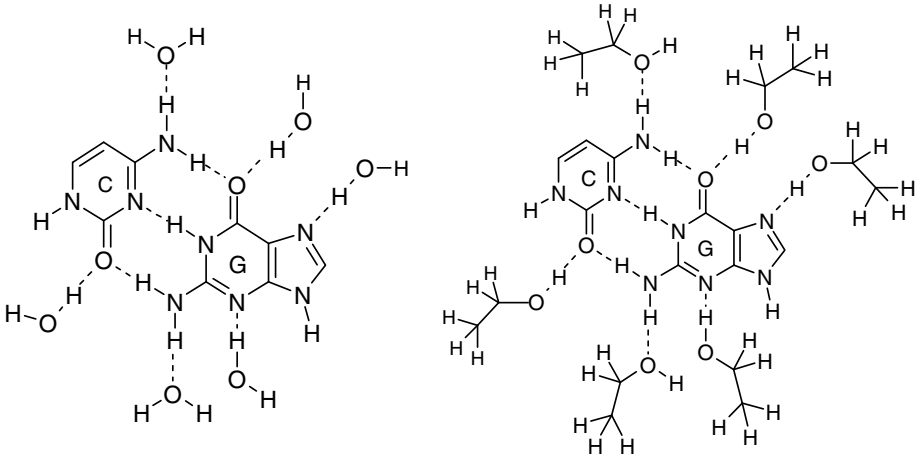
4 Results and Discussion

The results of calculations, Reichardt values for the environment polarity parameter (E_r) [17] and absorption bands in the UV spectrum in water (row 1) and ethanol (row 7) are presented in Table 1.

Table 1. Energy of activation ($\Delta E^{\#}$), energy of reaction (ΔE), tautomeric equilibrium constants $K_T(LL)$ and $K_T(AI)$, frequency of point mutation (ν_m), parameter of solvent polarity (E_T) and band of absorption in UV spectra (λ)

N	Environment	$\Delta E^{\#}$, kJ/mol	ΔE, kJ/mol	$K_T(LL)$	$K_T(AI)$	$\nu_m 10^{-2}$	$E_T[20]$	λ, nm
1	6w	35.0	24.5	0.103	0.106	1.09	1.000	260
2	5w+1eth	31.5	20.5	0.120	0.101	1.21		
3	4w+2eth	41.7	31.1	0.121	0.105	1.27		
4	3w+3eth	22.2	11.9	0.136	0.095	1.29		
5	2w+4eth	55.7	42.2	0.118	0.113	1.34		
6	1w+5eth	30.4	18.4	1.005	0.740	74.40		
7	6eth	25.0	-11.7	1.064	0.732	77.90	0.656	273

w-water; eth-ethanol.



a b

Fig. 1. Water (a) and ethanol (b) environment of the guanine-cytosine pair

In the first column, relative water and ethanol content in the environment is given. According to Löwdin[18], the point mutation frequency can be estimated by the product (multiplication) of concentrations of rare tautomeric forms of nucleotide bases. However, determination of these concentrations is connected with great experimental difficulties. Therefore, the mutation frequency was expressed by the multiplication of lactam-lactim KT(LL) and amino-imino KT(AI) tautomeric equilibrium constants since these constants are ratios of fractions of these tautomeric forms:

$$\nu_m = K_T^{LL} . K_T^{AI} \tag{1}$$

Activation energy, reaction energy, tautomeric equilibrium constant and mutation frequency values depend on the length of triads of hydrogen bonds between nucleotide bases (RNHO) [19]. In particular, equilibrium constants depend inversely on the length of the hydrogen bond triad.

$$K_T = \frac{k_1}{R_{NHO}} \tag{2}$$

On the other hand, using NMR spectrometry, it was shown that the polar solvent causes a decrease in KT [20,21].

$$K_T = \frac{k_2}{E_T} \tag{3}$$

From the comparison of formulas (2) and (3) we obtain that

$$E_T = k_3 R_{NHO} \tag{4}$$

where k3=k2/k1 are the proportionality coefficients.

Hence, mixing of ethanol with water leads to reduction of length of the hydrogen bonds between nucleotide bases, which facilitates the proton transfer.

Table shows that the decrease of the number of molecules of water and the increase of the number of molecules of ethanol significantly decreases the energy of activation of a protons transfer between GC nucleotide pair. Which in turn increase both the value of constant of the tautomeric equilibrium and value of relative frequency of mutation. In the case when one molecule of water and five molecules of ethanol (Table 1, row 6), and also 6 molecules of ethanol (Table 1, row 7) are located around the GC pair, the processes of a protons transfer become exothermic, that causes sharp increase of the relative frequency of mutation. Therefore the reduction of the environment polarity around the nucleotide base causes increase of DNA's tendency to mutation and denaturation. The solvatochromic effect mechanism can be explained by a specific action of solvent molecules on functional groups of cytosine and guanine involved in the proton transfer. This may be an inductive influence of the ethyl group of ethanol.

In particular, the electron-donor ethyl group ($\sigma 1$=-0.1) strengthens ethanol O-H bonds and thereby weakens O…H(N) and O…H(=O) bridges. As a result, the activation energy of the proton transfer of the cytosin amino group along the hydrogen bond decreases more than it could be expected in the case of full water surrounding. Hence, the decrease in the environment polarity (ET) increases the mutation frequency and the tendency of DNA to denaturation. This process can be described quantitatively by the solvatochromic effect, in particular, by the bathochromic shift of the absorption band of the DNA ethanol solution in the UV spectrum (Fig.2,3).

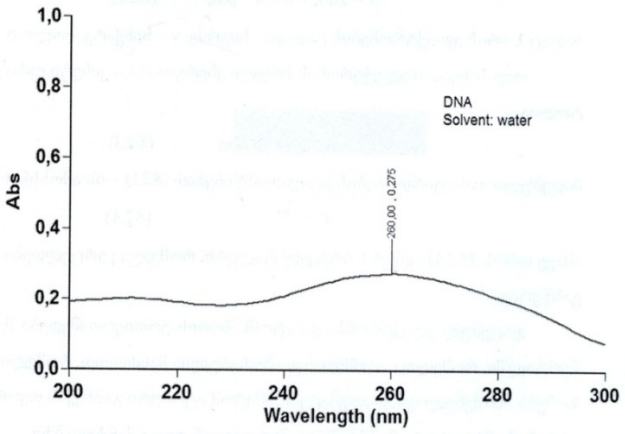


Fig. 2. UV spectra of DNA in water

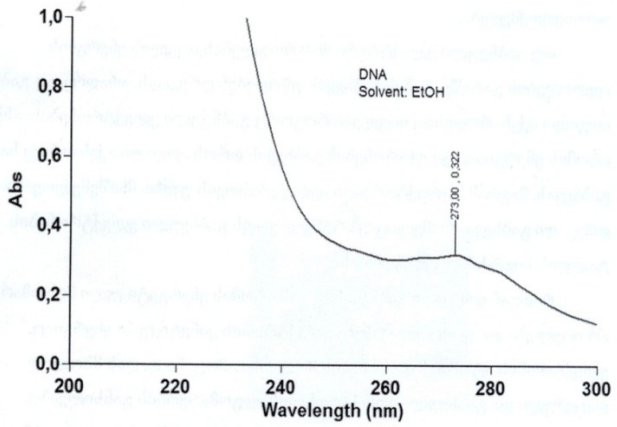


Fig. 3. UV spectra of DNA in ethanol

5 Conclusion

The results of data analysis of the quantum-chemical calculations and UV spectrum suggest that substitution of water molecules by ethanol molecules causes a decrease in energy and kinetic characteristics of the proton transfer between nucleotide bases. Hence, proposed the solvatochromic effect can serve for quantitative estimation of the influence of the environment polarity on the mutation and denaturation of DNA.

Acknowledgment. These investigations has been supported by European Commission FP7 project "High-Performance Computing Infrastructure for South-East Europe's Research Communities", Grant No: 261 499. We thank Prof. Ramaz Kvatadze for fruitful discussions and his assistance in obtaining access to the HP-SEE infrastructure.

References

1. Gorb, L., Podolyan, Y., Dziekonski, P., Sokolski, W., Leszczynski, J.: Double Proton Transfer in AT and GC Base Pairs. J. Am. Chem. Soc. 126, 10119–10129 (2004), doi:10.1021/ja049155n
2. Fogarasi, G.: Water-mediated tautomerization of Cytosine to the rare imino form. Chem., Phys. 349, 204–209 (2008), doi:Doi.org/101016/j.chemphys.2008.02.016
3. Ceron Carrasco, J.P., Requena, A., Michaux, C., Perpete, E.A., Jacquemin, D.: Effects of Hydration of the Proton Transfer Mechanism in the Adenine-Thymine Base Pair. J. Phys. Chem. A 113, 7892–7898 (2009), doi:10.1021/jp900782h)
4. Ceron Carrasco, J.P., Requena, A.J., Zuniga, J., Michaux, C., Perpete, E.A., Jacquemin, D.: Intermolecular Proton Transfer in Microhydrated GC Base Pairs: A New Mechanism for Spontaneous Mutation in DNA. J. Phys. Chem. A 113, 10549–10556 (2009), doi:10.1021/jp906551f

5. Cui, S., Yu, J., Kuhner, F., Schulten, K., Gaub, H.F.: Double Stranded DNA Dissociates into Sigle Strands When Dragged into a Poor Solvent. J. Am. Chem. Soc. 129, 14710–14716 (2007), doi:10.1021/ja74776c
6. Geidushek, E.P., Herskovits, T.: Nonaqueous Solvents of DNA. Reversible and irreversible denaturation in methanol. Arch. Biochem. Biophys. 95, 114–129 (1961), doi:org/10.1016/0003-9861(61) 90116-3
7. Sinanogly, O., Abdulnur, S.: Hydrophobing stacking of bases and the solvent denaturation of DNA. Photochem. and Photobiol. 3, 333–342 (1964), doi:10.1111/j.1751-1097.1964.tb08156.x)
8. Bonner, G., Klibanov, A.M.: Structural Stability of DNA in nonaqueous solvents. Biotech. Bioeng. 68, 339–344 (2000), doi:(10.1002/CSICI) 1097-0290(20000505)68
9. Arscott, P.G., Ma, C., Wennar, G.R., Bloomfield, V.A.: DNA condensation by cobalt hexamine in alcohol-water mixtures: Dielectric constant and other solvent effects. Biopolymers 36, 345–364 (1995), doi:10.1002/bip.360360309
10. Feig, M., Pettitt, B.M.: A molecular simulation picture of DNA hydration around A-and B-DNA. Biopolymers 48, 199–209 (1998), doi:10.1002/(SICI)1097-0282(1998)48
11. Chalikyan, T.V., Sarvazyan, A.P., Plum, G.E., Breshauer, K.J.: Influence of Base composition, base sequence. and duplex structure on DNA hydration. Biochemistry 33, 2394–2401 (1994), doi:10.1021/bi00175a007
12. Kohn, W., Becke, A.D., Parr, R.G.: Density Functional Theory of Electronic Structure. J. Phys. Chem. 100, 12974–12980 (1996), doi:10.1021/jp960669l
13. Laikov, D.N., Ustynyuk, Y.A.: Priroda_04: a quantum-chemical program suite. New possibilities in the study of molecular systems with the application of parallal computing. Russ. Chem. Bull., Int. Edn. 54, 820–826 (2005)
14. Perdew, J.P., Burke, K., Ernzerhof, M.: Generalized Gradient Approximation Made Simple. Phys. Rev. Lett. 77, 3865–3868 (1996), doi:10.1103/PhysRev.lett.77.3865
15. Adamo, C., Barone, V.J.: Physically motified density functional with improved performances: The modified Perdew-Burke-Ernzerhof model. J. Chem. Phys. 116, 5933–5941 (2002), doi.org./101063/1.1458927
16. Becke, A.D.: Density-functional exchange-energy approximation with correct asymptotic behavior. Phys. Rev. A38, 3098–3100 (1988), doi:10.1103/PhysRevA38.3098
17. Reichardt, C.: Solvatochromic dyes as solvent polarity indicators. Chem. Rev. 94, 2319–2358 (1994), doi:10.1021/cr00032a006
18. Lowdin, P.: Some Aspects on the Biological Problems of heredity, Mutations, Agingand Tumors in View of the quantum theory of the DNA molecule. Advances in Quantum Chemistry 2, 213–360 (1966), doi:org/10.1016/S0065-3276(08)60069-6
19. Kereselidze, J.A., Zarqua, T., S. Kikalishvili, T.J., Churgulia, E.J.M.C., Makaridze, M.C.: Somenew views on the tautomerisation mechanism. Russ. Chem. Rev. 71, 993–1003 (2002)
20. Rogers, M.T., Burdett, J.T.: Keto-enoltautomerism in β-dicarbonyls studied by nuclear magnetic resonance spectroscopy. J. Am. Chem. Soc. 86, 2105–2109 (1964), doi:10.1021/ja01065a003
21. Prezhdo, V.V., Khimenko, N.L., Surov, Y.N.: Influence of the solvent on tautomeric transformation of acetoacetic ester. Ukr. Khim. Zh. 52, 57–63 (1986)

Dynamic Features of Complex Systems: A Molecular Simulation Study

Armen Poghosyan[1], Levon Arsenyan[1], and Hrachya Astsatryan[2]

[1] International Scientific Educational Center of the National Academy of Sciences of the Republic of Armenia, 24D, M. Baghramyan ave., Yerevan 0019, Armenia
sicnas@sci.am, levon.arsenyan@gmail.com
[2] Institute for Informatics and Automation Problems of the National Academy of Sciences of the Republic of Armenia, 1, P. Sevak str., Yerevan 0014, Armenia
hrach@sci.am

Abstract. The main aim of the article is the molecular simulation study and detailed analysis of surfactant molecules of complex micellar systems [1-2] consist of long hydrocarbon chain surfactant. The GROMACS software package [3] designed for high-performance simulations of large complex systems is used for the simulations. The IBM BlueGene/P supercomputer [4] at Bulgarian National Centre for Supercomputing Applications, with 8,192 processor cores connected by multiple high-performance networks, enables to investigate a completely new class of problems. The initially random distributed surfactant molecules in aqueous solute hydration have been simulated using GROMOS united atom force field [5]. An extensive series of short benchmarks run for timing purposes with different number of cores show that the studied system achieves good scalability (0.5ns per day) in case of using up to 512 processor cores. Further increasing the numbers of cores (for instance, 1024 cores) does not lead any significant increase. In spite of the limitation of number of simulations, the qualitative statistical data gave some interesting results, which indicates that long hydrocarbon chain surfactant self-assemble into small oligomers since 50ns of simulations, meanwhile in our previous study with surfactant rich content shows that 43ns is enough for self-assembling of spherical micelle.

Keywords: Parallel Molecular Dynamics, Surfactants, Micelle, GROMACS, BlueGene.

1 Introduction

The amphiphilic molecules, consist of hydrophilic headgroup and hydrophobic tails, self-organize various structures (micelles, lamellas, etc) and plays an important role in many fields [6]. In parallel with the real experiments [7], a number of computational experiments have been carried out in order to study the micellization phenomena of surfactants. However, the authors mainly use preassembled micelles as an initial configuration, considering that the self-organization process takes a very long time and as a benchmarking surfactant, the sodium dodecylsulfate (SDS) is used. As a result of simulations, it is important to mention that surfactant concentrations much higher than

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_14, © Springer International Publishing Switzerland 2014

their critical micellar concentrations (CMC). The simulations near the CMC require large-scale computational resources.

The main aim of the article is to estimate and study the behavior and dynamics of aggregation kinetics of long chain sodium pentadecyl sulfonate (SPDS) molecule close to CMC concentration. As the system size reached more than two million atoms, mainly the computational resources of the Bulgarian BlueGene/P supercomputer are used for the simulations and getting the scalability of the system, which is affected by many factors [8].

2 Construction and Simulation Data

The investigation of SPDS surfactant has been performed using the model of SPDS molecule [9]. By means of replication of 128 molecules of SPDS are received, which are located randomly in the unit box with the dimensions of 36x36x20nm. Counterions are set at 0.3 nm from SPDS headgroups. The received systems, consist of about 2.6 million atoms, have been realized using GROMACS standard module genbox. Eventually, 5000 steps of energy minimization have been performed by steepest descent algorithm. The system has been simulated in isothermal-isochoric (NVT) ensemble using SPDS force field parameters generated by Dundee server [10] and standard SPC water models [11]. The integration time step is set to 2 fs. Atom bonds are regulated by LINCS constrain algorithm [12]. PME method is used for long-range electrostatic interactions [13] and the van der Waals interactions are truncated at1.2 nm with the short-range neighbor list distance at 1 nm. Isotropic pressure with $\tau = 0.1\text{ps}$ time constant is coupled using Berendsen barostat and the constant temperature T=300K are maintained by V-rescale algorithm. Three-dimensional periodic boundary conditions have been applied. The coordinates and velocities are saved every 1ns and the VMD package [14] is used for the visualization of the system.

Overall 100ns of parallel molecular dynamics (MD) simulation in NPT ensemble, which is more time consuming than in the canonical (NVT) ensemble, has been carried out using computational resources (from 256 to 1024 cores) of Blue Gene/P supercomputer.

3 Results and Discussions

Several benchmarks have been carried out [15] in order to estimate the speedup using the computational resource of the Blue Gene/P supercomputer. The obtained data show about 0.5ns per day achieved by using 512 processor cores. The further increase of cores up to 1024 do not lead any significant speedup and therefore 512 processor cores are chosen for the final simulations.

In order to visualize the structural changes in more detail and estimate the oligomer size, we provide snapshots of the system at the start point, 50ns and 100ns run points. In Figure 1(a,b,c) the cross-sectional view is shown. The provided snapshots indicate that he long hydrocarbon chain surfactant self-assemble into small oligomers since 50ns of simulation run (Figure 1b), where the maximum size of aggregation is about 6 molecules and we track a few amount of this kind of aggregations (about 6-7).

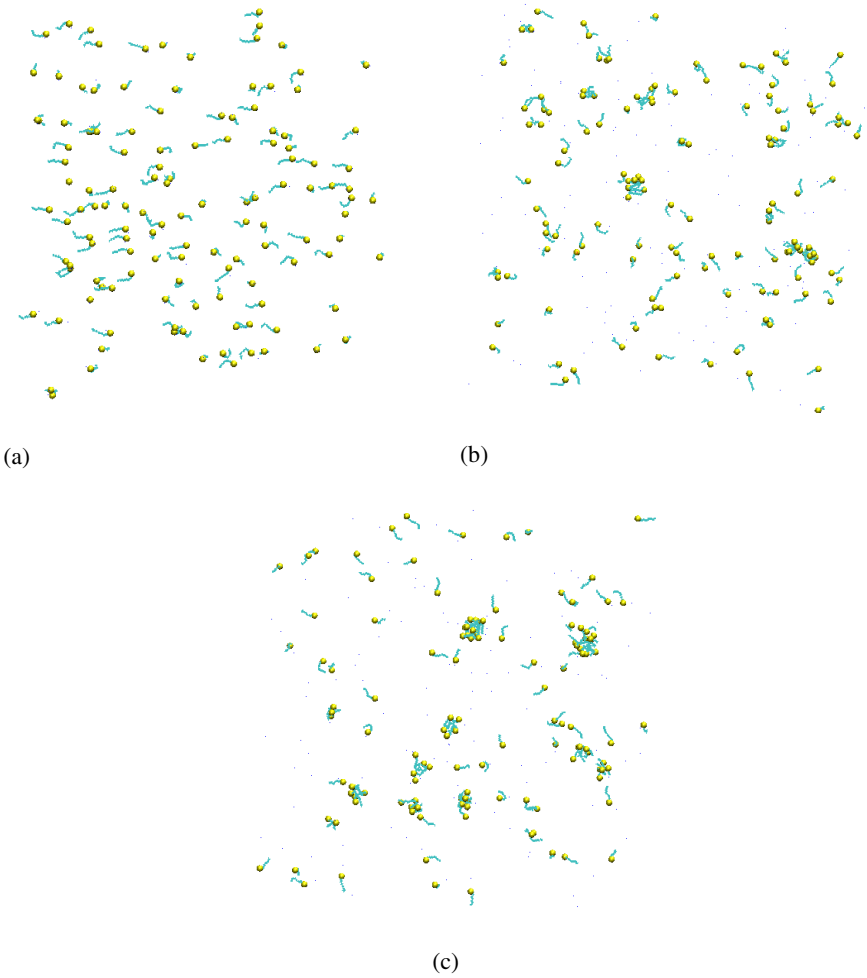


Fig. 1. (a, b,c). Three stages (0ns, 50ns and 100ns) of surfactant self-organization process. The water molecules have been omitted for clarity. The sulfur drawn as VDW sphere and other components are given as stick.

It is known that the self-assembling process takes longer for surfactants with long hydrophobic chains due to slower self-diffusion [16]. However, it is also stated that the surfactant rich content (higher concentrations) shows that about 43ns is enough for self-assembling of spherical micelle. Comparing two concentrations (near CMC and higher concentration) shows that the same aggregation behavior for surfactant rich content was seen at about 6-7ns run point. The further simulations up to 100ns (Figure 1c) show that there is an increase of aggregation size and amount reaching up to 10 oligomers with 10-12 surfactant molecules.

Currently, we continue the simulations to investigate the self-assembling process of huge complex systems.

4 Conclusion

The MD simulations of long hydrophobic chain sulfonate/water system have been studied at different concentration. The self-organization behavior of the sodium pentadecyl sulfonate surfactant has been investigated by carrying out simulations using a random distribution as an initial configuration.

We conclude that the self-assembling micellization process takes longer for surfactants with long hydrophobic chains. It is planned to accelerate further simulations using GPUs, as a standard tool for high performance computing and application acceleration, as well as to develop a grid-enabled Web portal [17] for the setup and execution of MD simulations for complex systems.

Acknowledgements. This work was supported in part by the European Commission under EU FP7 project HP-SEE (N261499).

References

[1] Wang, Y., Wallace, J.A., Koenig, P.H., Shen, J.K.: Molecular dynamics simulations of ionic and nonionic surfactant micelles with a generalized Born implicit-solvent model. Journal of Computational Chemistry 32(11), 2348–2358 (2011)
[2] Jorge, M.: Structure of cationic surfactant micelles from molecular simulations of self-assembly. Journal of Molecular Structure 946, 88–93 (2010)
[3] Berendsen, H.J.C., van der Spoel, D., van Drunen, R.: GROMACS: A message-passing parallel molecular dynamics implementation. Computer Physics Communications 91(1-3), 43–56 (1995)
[4] Team, IBM Blue Gene, Overview of the IBM Blue Gene/P project, IBM Journal of Research and Development 52.1/2,: 199-220 (2008)
[5] van Gunsteren, W.F., Berendsen, H.J.C.: Groningen Molecular Simulation (GRO-MOS) Library Manual, BIOMOS b.v., Groningen (1987)
[6] Shah, D.O.: Micelles, Microemulsions and Monolayers: Science and Technology. Marcel Dekker, New York (1998)
[7] Shahinyan, A.A., Khanamiryan, L.A., Aivazyan, O.M., Nalbandyan, Y.E., Grigor-yan, J.D.: Influence of water-soluble polymers on the properties on the aqueous micel-lar solutions of an anionic surfactant and on the process of micellar-emulsion polymerization of styrene. Colloid Journal 49, 521–529 (1987)
[8] Affinito, F., Emerson, A., et al.: Performance Analysis and Petascaling Enabling of GROMACS, PRACE white paper
[9] Arsenyan, L.H.: Some problems of molecular dynamics study of sodium pentadecyl-sulfonate/water lyotropic system. New Electronic Journal of Natural Sciences 1, 4–6 (2009)
[10] Schuettelkopf, A.W., van Aalten, D.M.F.: PRODRG – a tool for high-throughput crystallography of protein-ligand complexes. ActaCrystallogr. D 60, 1355–1363 (2004)
[11] Berendsen, H.J.C., Postma, J.P.M., van Gunsteren, W.F., Hermans, J.: Intermolecular Forces, pp. 331–342. Reidel, Dordrecht (1981)
[12] Hess, B., Bekker, H., Berendsen, H.J.C., Fraaije, J.: LINCS: A Linear Constraint Solver for Molecular Simulations. Journal of Computational Chemistry 18, 1463–1472 (1997)

[13] Darden, T., York, D., Pedersen, L.: Particle mesh Ewald: An N log (N) method for Ewald sums in large systems. Journal of Chemutational Physics 98, 10089–10092 (1993)
[14] Humphrey, W., Dalke, A., Schulten, K.: VMD - Visual Molecular Dynamics. Journal of Molec. Graphics 14, 33–38 (1996)
[15] Poghosyan, A.H., Arsenyan, L.H., Astsatryan, H.V.: Comparative NAMD Bench-marking of Complex System on Bulgarian BlueGene/P. In: IEEE Proceedings of 35th International Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, Croatia, pp. 319–321, ISBN: 978-1-4673-2577-6
[16] Ben-Shaul, A., Gelbert, W.M.: Micelles, Membranes, Microemulsions and Monolayer. Springer, New York (1994) ISBN: 978-1-4613-8391-8
[17] Astsatryan, H., et al.: Services Enabling Large-Scale Linear Systems of Equations and Algorithms based on Integrated P-Grade Portlal. Springer Journal of Grid Computing (2013), doi:10.1007/s10723-013-9254-7

Using Adaptive Mesh Refinement Strategy for Numerical Solving of Gas Dynamics Problems on Multicore Computers

Boris Rybakin[1], Peter Bogatencov[2], Grigore Secrieru[2], and Nicolai Iluha[2]

[1] MSU, Lenin Mountains, Building 1; SRISA RAS, Moscow, Russia
rybakin@vip.niisi.ru
[2] Institute of Mathematics and Computer Science, Academy of Sciences of Moldova
{bogatencov,secrieru,iluha}@renam.md

Abstract. This paper describes an algorithm and a program for solving multidimensional problems represented by differential equations with partial derivatives adopted for using SEE regional HPC resources. The algorithm is based on the AMR method - Adaptive Mesh Refinement of computational grid. Utilization of AMR method can significantly improve resolution of difference grid in areas of high interest, and also to accelerate processes of multidimensional problem solving.

Keywords: multidimensional modelling, AMR method, parallel algorithms, OpenMP.

1 Introduction

New approaches applied to science exponentially increase computational requirement in order to realistically describe and solve real-world problems, numerical simulations become more detailed, experimental sciences use more complicated instruments to make precise measurements; and shift from individuals-based science towards a collaborative research model now starts to dominate. Mathematical modelling forms a solid theoretical and applied basis in describing, simulating and studying complex problems. Parallel applications that require utilization of high-performance computers ensure necessary accuracy of mathematical models. Although new perfect high performance computational installations become available for researchers, in many cases their resources are not sufficient for solving many practical problems. This may seem a paradox, but the fact of the matter is the more computational resources become available, the larger problems scientists and engineers want to solve. Thus despite the constant computer power growth, finding of effective algorithms and approaches for adequate models realization remain the key for solving of complex practical problems. One of the methods that allows developing optimized applications and speeding up the process of complicated models execution is the method based on adaptive refinement of computational mesh - AMR (Adaptive Mesh Refinement) method [1].

In this paper we focus an solving two- and three- dimensional tasks of gas dynamics that are of practical interest. These solutions can be applied to many

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_15, © Springer International Publishing Switzerland 2014

present-day problems, such as calculation of the aerodynamics of aircraft, the calculations of the air flow of cars, a large number of other similar problems of mathematical modelling - the calculation of the flow of blood through the vessels, the heart valves, etc. In recent years effective high-resolution difference schemes have been developed, which allow mathematical modelling of the processes mentioned above. But making three-dimensional calculations for high definition grids requires large computational resources.

In all these cases, at the beginning of the problem we define a way to highlight areas in which we need to construct a grid, then the program builds a sequence of grids and makes a decision on them. During calculations in the computational area there are domains with large gradients of the parameters - such as temperature, pressure, density, and several others. These areas are contiguous with areas with a smooth behavior of the investigated functions. Therefore, to reduce the requirements for computing resources a detailed grid with small mesh sizes can be created only in areas of high gradients. An approach to creation of such area - adaptive mesh refinement will-significantly clarify definition of multi-dimensional flows features [2].

We will consider a computer simulation of gravitational collapse of stars with masses ranging from 7 to 70 solar masses [3]. This process leads to formation of a supernova and requires using of non-oscillating schemes of high resolution. Schemes used for modelling should be with maximum accuracy to reproduce the behaviour of matter in the vicinity of discontinuities, accurately describe the emissions of substances and small perturbations far away from the fronts of shock waves [4]. To address these conflicting requirements, finite difference schemes are used, which combine high resolution in areas of small disturbances and smooth monotony in the areas of strong discontinuities.

2 Methods

2.1 Formulation of the Problem

The system of equations of gravitational gas dynamics, which describes the process of collapse of a star is

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial v_i}(\rho v_i) = 0$$

$$\frac{\partial(\rho v_i)}{\partial t} + \frac{\partial}{\partial v_i}(\rho v_i v_j + P\delta_{ij}) = -\rho \frac{\partial \phi}{\partial x_i} \tag{1}$$

$$\frac{\partial e}{\partial t} + \frac{\partial}{\partial x_i}[(e+P)v_i] = -\rho v_i \frac{\partial \phi}{\partial x_i} \ .$$

In these equations, the value of the gravitational potential ϕ is determined from the Poisson equation

$$\Delta\phi = 4\pi G\rho \ . \tag{2}$$

The equation of state is used in the form of:

$$e = 1/2\rho v^2 + \varepsilon \ .$$

In the above equations: ρ - density, v - velocity, P - pressure, ε - specific internal energy, e - full energy, t - time, x_i - spatial coordinates, $G = 6,67 \times 10^{-11}\mathrm{m}^3/(\mathrm{kg} \cdot s^2)$ - constant gravitational potential.

The parameters of gas collapsing star are determined from the numerical solution of equations (1) and (2). It should be taken into account that density of a collapsing star changes by many orders - from 10^{14} kg/m^3 at the center of a neutron star to the density of a rarefied gas on the boundary of the stellar envelope [3]. It is therefore necessary to create a grid, the size of which depends on the density, that is, the cells in the center should have minimum size, and should increase with distance from the center.

2.2 AMR Method

Calculations method based on the use of AMR hierarchical grid cells can significantly improve quality of the calculations in various fields of science and engineering calculations. In addition, most of applications that use AMR is well parallelized on supercomputers. We use the programming language Fortran 90. Program based on AMR technology uses object-oriented approach, which is available in Fortran 90 [2].

Such kind of the grid can be constructed by using the mechanism of AMR [5]. After the initial grid constructing, all cells of the same level of decomposition should be united into rectangular patches. Thus we get a hierarchical grid for solving differential equations in partial derivatives. The hierarchy of this grid is very complex and uses different levels of grid refinement from the coarsest level ($L = 0$) to the finest level ($L = L_{max}$). Each level is a collection of patches of various sizes. Also the dividing of the computational area changes with time. The coarsest level ($L = 0$) is a rectangular box, which contains all the other levels. In addition, it is assumed that all other levels are also strictly nested.

Each level is represented by a structure [2,8], which contains all the information about this level. When creating a new level, a new structure is generated, that is being filled with the necessary information, including the boundary conditions. Each transition from the coarse level (m) to the "finer" one ($m + 1$) have to be correctly calculated when moving to a more accurate grid, and vice versa from the level of ($m + 1$) to level m, when moving to the initial grid of the upper level. Subroutines *CoarseToFine* and *FineToCoarse* are used for this in the elaborated application. At every level the calculations are carried out with a few number of algorithms - Jacobi simple iteration, method of successive over-relaxation (SOR), conjugate gradient method. The choice of algorithm depends on which computer the application is calculated. Simple iteration method is used on supercomputers that use GPU (Graphical Processor Unit). Jacobi method allows good parallelization when it is calculated on GPUs. Methods with better convergence SOR and conjugate gradient are used for computers without GPUs and they are well parallelized using OpenMP programming technology [6].

2.3 Calculation Algorithm

The solution of the three-dimensional Poisson equation (2) with given initial and boundary conditions is described below. This solution has been tested on 5 levels of AMR nesting. The test No 1 from [7] is investigated here below. In this test a homogeneous sphere of radius R and density ρ for the equation for the gravitational potential (4) is considered:

$$\phi(r) = \begin{cases} 2\pi G\rho_0(R^2 - r^2/3) & \text{if} \quad r \le R \\ \frac{4}{3}\pi G\rho_0 R^3/r & \text{if} \quad r > R \end{cases} \tag{3}$$

$$\nabla\phi(r) = \begin{cases} -\frac{4}{3}\pi G\rho_0 r & \text{if} \quad r \le R \\ -\frac{4}{3}\pi G\rho_0 R^3/r^2 & \text{if} \quad r > R\ . \end{cases} \tag{4}$$

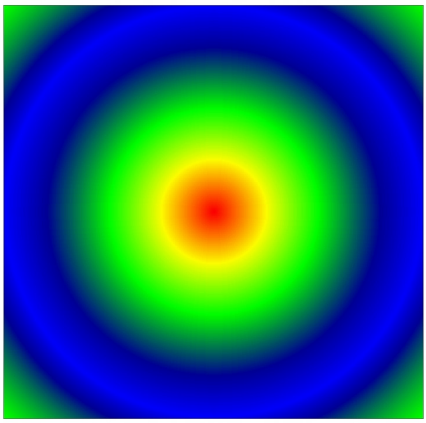
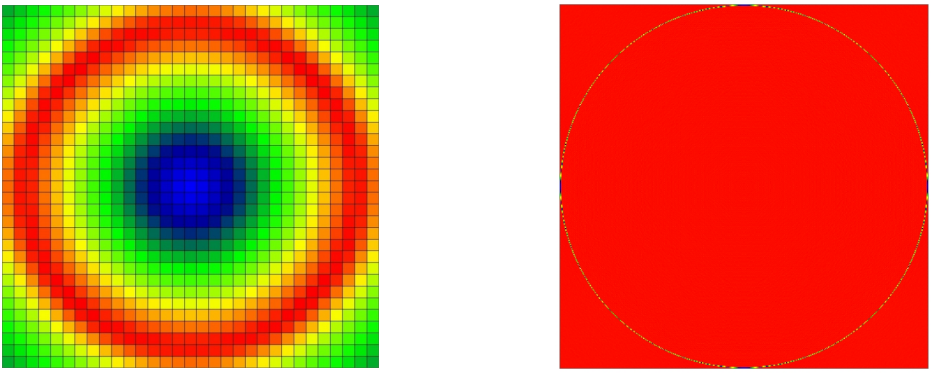


Fig. 1. Numerical solution for the fourth level of AMR - 2-D image

The computational domain is filled with values ϕ from (3) in the sphere of radius R, outside the sphere the values from (3) are specified - bottom line. Then the values of the gravitational potential are calculated in the three-dimensional formulation [8]. The numerical solution was calculated for AMR hierarchy levels from one to five.

The numerical results present two-dimensional cross-section of the X-axis in Fig. 1. From the analysis of the images shown in Fig. 1 it should be noted a good coincidence of the analytical and numerical solutions. Figure 2 (a) and (b) provides a graphical expression of the error value as the difference between the analytical and numerical solutions for the 32x32 grid's dimension and two levels of the grid. Maximum value of the error is equal to $5{,}6 \cdot 10^{-4}$. It should be noted that we have regular, not random errors. It is tied with a nature of the functions themselves. To calculate the first derivative we used finite differences of second-order accuracy.



(a) (b)

Fig. 2. Graphic representation of the error: for the grid of 32x32 and two levels (a) and for the grid of 512x512 and five levels (b)

Figure 2 (b) shows the image of the error, which is the difference between analytical and numerical solutions for the grid size of 512x512 and 5-level refinement of the grid. It should be mentioned the significant refinement of result and consequent decreasing of the error by four orders of magnitude. These results show the importance of applying the AMR method for raising accuracy of the solutions of multidimensional partial differential equations.

3 Results

3.1 AMR_PAR 64-Bit Application

The first version of the application was developed using MS Visual Studio 2010 (for OS MS Windows) [10]. However, the specifics of applications, prepared for running using regional HPC resources provided by HP-SEE project infrastructure, is that they should not have an interactive mode. Thus, the absence of a graphical interface, the use of standard functions and libraries simplify problems of portability and allow porting of the application by reducing its adaptation to a simple recompilation of the source code.

In this case, one of the ways to avoid errors when the application is ported to the regional computational resources, is using compilers, produced by one developer for both Windows and Linux. For example, there are versions of Intel Parallel Studio XE for Windows and Linux.

The first results were obtained by using the version of the application developed for MS Visual Studio 2010 (OS MS Windows) [3] on PC with processor Intel Core I7 920, 2667 GHz. Calculations were performed using option *collapse (n)*, and without it. Option *collapse* is used for apportionment of the iterations in strongly nested loops. In Fig. 3 there is a graph of acceleration depending on

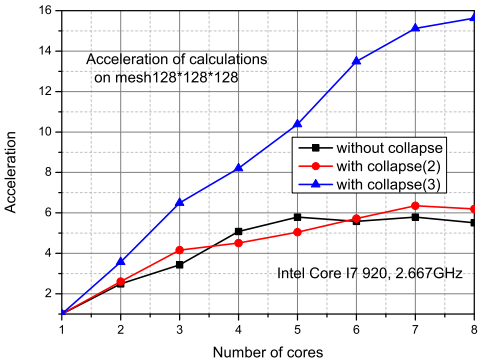


Fig. 3. Acceleration of calculations. MS Windows version Intel I7 920

the number of processor cores for computational grid 128x128x128. The graph with triangles shows acceleration achieved using options *collapse* on three nested loops, graphs with circles show acceleration with *collapse* on 2 cycles, and squares show acceleration without *collapse*.

Note that the use of this directive for three strongly nested loops greatly speeds up the program. This is because for all cycles a common space of iterations is formed, which is divided between all threads [9].

The same job was performed on a dual-node cluster with Intel Xeon 5560 processors, each has 8 cores and does not support HyperThreating. Figure 4 shows a graph of the acceleration for grid dimension 128*128*128 and 256*256*256. It should be noted a significant drop in performance in comparison with the previous calculations. This can be explained with the lower clock speed of the processor and other size of cash memory.

The AMR_PAR application testing was performed initially in OpenMP mode on Microsoft Windows Compute Cluster 2003 in the Institute of Mathematics and Computer Science of the Academy of Science of Moldova. In this cluster were held debug and test calculations, which were then used to be calculated on a more powerful computers.

3.2 Adopting AMR_PAR Application to the Regional HPC Resources

After calculation of the necessary amount of RAM for grid dimensions up to 2048x2048x2048 cells, as home cluster for the application porting the SGI UltraViolet 1000 supercomputer at the National Information Infrastructure Development Institute, located in Pecs City, Hungary (SMP, 1152 cores and 6057 GB RAM) was proposed. For small grids (up to 384x384x384 cells), we used resources of HPCG cluster located at the Institute of Information and Communication Technologies of Bulgarian Academy of Sciences.

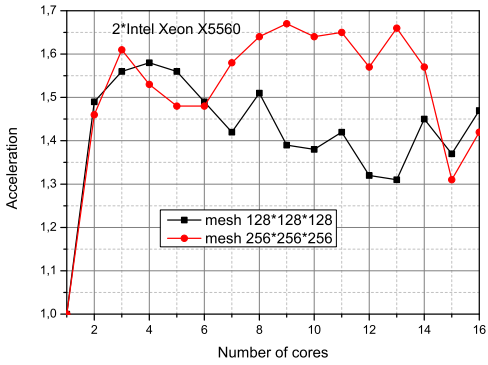


Fig. 4. Acceleration of calculations. Two Intel Xeon 5560.

In order to save computational resources in case of multiple execution of the application it is important to predict the optimal number of cores for calculations of arrays with different dimensions and different levels of nesting. The modified version of application was compiled and tested on HPCG cluster in Bulgaria (Intel XeonX5560, 2,8 Ghz, 24 Gb RAM). For arrays of 128x128x128 dimension with five levels of nesting optimal number of cores for calculation is about 4–5. Increasing the number of cores does not reduce wall-clock time of calculations, but the processing time increases dramatically.

4 Discussion and Conclusions

On the base of run time analysis we calculated various versions of the application requirements of computational resources for the current OpenMP version of AMR_PAR application:

- For HPCG cluster located at the Institute of Information and Communication Technologies of Bulgarian Academy of Sciences maximum grid dimension for five layers is 384x384x384, approximate time of calculations - 5 hours, optimal number of cores - 8.
- Calculations for 5-7 levels and grid dimensions sizes more than 384x384x384 require up to 3 Tb of RAM.

The obtained results allow to carry out calculations on modelling of supernova explosions and formation of stars from molecular clouds. This program is now implementing being transformed for integration into the software system for the calculation of three-dimensional gas-dynamic flows, to perform calculations.

4.1 Future Work

For further optimization of AMR_PAR application, we plan collecting statistics of calculations' acceleration dependencies from different number of cores - up to

32 (or more). It is necessary to produce investigations to find optimal number of cores for fastest calculations for large-scale grid dimensions. As a result of this research we plan to modify the application to use OpenMP more effectively.

Next step is to run application using HP-SEE regional resources for large-scale grid dimensions - up to 2048x2048x2048 - and 5-7 layers. After obtaining results of the modified application execution, it will be possible to make new benchmarking (due to long time of forecast calculations) and propose new recommendations for application optimization. The updated application also will allow describing and visualization of results in 2-D images and 3-D models.

Acknowledgements. This work is supported by European Commission project High-Performance Computing Infrastructure for South East Europe's Research Communities (HP-SEE) under contract number 261499.

References

1. Berger, M.J., Oliger, J.: Adaptive mesh refinement for hypervelocity partial differential equations. Journal of Comp. Phys., Elsevier 53, 482–512 (1984)
2. Balsara, D.S., Norton, C.D.: Highly parallel structure adaptive mesh refinement using parallel language-based approaches. Elsevier, Parallel Computing 27, 37–70 (2001)
3. Bisnovatyi-Kogan, G.S., Moiseenko, S.G., Rybakin, B.P., Secrieru, G.V.: Modeling of explosive magneto rotational phenomena: from 2D to 3D. Buletinul ASM, Mathematica N.3(55), 55–63 (2007)
4. Rybakin, B.P.: Parallel three-dimensional TVD scheme for solving problems of gravitational gas dynamics. In: Parallel Computing Technologies (PaCT 2009), Nizhny Novgorod, pp. 673–679 (2009) (in Russian)
5. Klein, R.I.: Star formation with 3-D adaptive mesh refinement: the collapse and fragmentation of molecular clouds. Journal of Computational and Applied Mathematics 109, 123–152 (1999)
6. Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J.: Parallel Programming in OpenMP. Morgan Kaufmann (2000) ISBN 1-55860-671-8
7. Stone James, M., Norman Michael, L.: ZEUS-2D: A Radiation Magneto-hydroynamics Code for Astrophysical Flows in Two Space Dimensions I. The Hydrodynamic Algorithms and Tests. The Astrophysical Journal-Supplement Series 80, 753–790 (1992)
8. Rybakin, B.P.: Modeling of III-D Problems of Gas Dynamics on Multiprocessing Computers and GPU. Elsevier, Computers & Fluids (January 31, 2012), doi:10.1016/j.compfluid.2012.01.016
9. Rybakin, B.P.: Numerical methods for multiprocessor computers. Moldova State University, Center for Education and Research in Mathematics and Computer Science, Chişinău (2008) ISBN 978-9975-70-700-8
10. Iliuha, N., Altuhov, A., Bogatencov, P., Secrieru, G., Golubev, A.H.-S.: Project - Providing Access to the Regional High Performance Computing Infrastructure. In: Proceedings IIS "International Workshop on Intelligent Information Systems", Chişinău, September 13-14, pp. 183–186 (2011) ISBN 978-9975-4237-0-0

Number Theory Algorithms on GPU Clusters

Emanouil Atanassov[1], Dobromir Georgiev[1], and Nikolai Manev[2]

[1] Institute of Information and Communication Technologies, BAS
Acad. G. Bonchev St.,Bl.25A, 1113 Sofia, Bulgaria
{emanouil,dobromir}@parallel.bas.bg
[2] IMI - BAS and VSU"L.Karavelov"
Acad. G. Bonchev St., Bl.8, 1113 Sofia, Bulgaria
nlmanev@math.bas.bg

Abstract. Many algorithms from Number Theory and their implementation in software are of high practical importance, since they are the building primitives of many protocols for data encryption and authentication of Internet connections. Number theory algorithms are also the basic part of cryptanalytic procedures. Many of these algorithms can be parallelized in a natural way. In this paper we describe our efforts to develop a software package that implements various Number Theory algorithms on GPU clusters and in partial our implementations of integer factorization using NVIDIA CUDA on clusters equipped with NVIDIA GPUs. Also we report results of our experiments regarding the performance of our implementation.

Keywords: integer factorization, GPU, CUDA, MPI.

1 Introduction

The problem of integer factorization is of high practical importance because of the widespread use of public key cryptosystems for encryption and authentication of Internet connections. The most used recently methods of integer factorization are the number field sieve and the elliptic-curve (ECM) methods. The latter was introduced by Lenstra in [10]. It can be considered as a generalization of Pollard's $p-1$ and Williams' $p+1$ methods. In the following years much efforts were spent to improve the ECM by proposing more suitable curves and forms of group operations on curve points. The reader can find a detailed description of the development of the ECM and rich bibliography in [14]. We will only mention the eminent Montgomery's paper [11], where the Montgomery's form of elliptic curves is introduced. In the passed decade new curves and corresponding group laws have been proposed (e.g. [9], [2], [4], see also [1]). Substantial efforts were dedicated to speed the algorithms using hyperelliptic curves, too.

The continuous interest in the ECM is due to its main advantage - it enables many curves to be used in parallel. That is why the ECM integer factorization is amenable to implementation on multiprocessor and multi-core computer architectures.

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_16, © Springer International Publishing Switzerland 2014

One of the most important recent developments in the area of high performance computing is the increasing use of Graphics processing units (GPUs) for general purpose computing. The main reason for this is the high efficiency of GPUs in terms of cost, space and energy usage. In order to make use of the advantages of GPU computing, one has to design the algorithms in a way that exploits the massive parallelism of the GPUs, where thousands of threads should run in parallel at clock frequencies, similar to that of CPUs. It is well established that the main algorithms, relevant to cryptography, are amenable for efficient implementation on GPUs (see, e.g., [5] [6]).

Since the computational power of a single server, equipped with GPU cards, may not be sufficient for the more challenging problems, one has to consider clusters of such machines, interconnected with Ethernet or Infiniband. The most popular paradigm for parallel computing in such environment is the Message Passing Interface (MPI) [8], [12].

2 Preliminaries

Let N be an integer. Consider elliptic curves of degree 3 in the projective plane over the ring $\mathbb{Z}_N$ of integers modulo N. Lenstra's method uses curves presented in short Weierstrass form

$$y^2z = x^3 + axz^2 + bz^3, \tag{1}$$

where $a, b \in \mathbb{Z}_N$, such that $4a^3 + 27b^2$ is invertible in $\mathbb{Z}_N$, $(N, 6) = 1$.

The Montgomery form [11] of curves is given by

$$BY^2Z = X^3 + AX^2Z + XZ^2, \tag{2}$$

where $A, B \in \mathbb{Z}_N$, such that $B(A^2 - 4)$ is an invertible element of $\mathbb{Z}_N$. The points $(X : Y : Z)$ on the curve form an additive abelian group with neutral element $\mathcal{O} = (0 : 1 : 0)$, $-Q = (X : -Y : Z)$ as the additive inverse of $Q = (X : Y : Z)$, the point group operation described as follows. In order to calculate $nP = (X_n : Y_n : Z_n)$ we start with $P = (X_1 : Y_1 : Z_1)$ and use the formulas below. Note that we need only two coordinates: X and Z. That is why the Montgomery form of curves is preferred for computer factorization.

Doubling

$$d = \frac{A+2}{4};\ s = (X_n + Z_n)^2,\ r = (X_n - Z_n)^2,\ t = d(s - r) + r,$$

$$X_{2n} = sr, \quad Z_{2n} = (s - r)t.$$

Adding (To calculate $(m + n)P$ we need mP, nP, and $(m - n)P$.):

$$u = (X_m + Z_m)(X_n - Z_n),\ v = (X_m - Z_m)(X_n + Z_n),$$

$$w = (u + v)^2,\ t = (u - v)^2,$$

$$X_{m+n} = Z_{m-n}w, \qquad Z_{m+n} = X_{m-n}t.$$

The additive inverse of $Q = (X : Y : Z)$ is the point $-Q = (X : -Y : Z)$ but there is no distinction between these points in the above formulas.

In our implementation we use only addition with $m - n = 1$, that is $X_{m-n} = X_1$ and $Z_{m-n} = Z_1$.

The factorization algorithm consists of two phases.

The First Phase

The point $Q = sP$ is computed, where s is the product of all powers of primes $p^k \leq B_1$. Then the coordinates X_Q and Z_Q are tested for having a nontrivial common divisor with N.

Usually the procedure of computing sP is the following. Let $s = (b_k b_{k-1} \dots b_1)_2$ have k bits in binary format, where the first bit is the least significant bit. Starting with b_{k-1} every bit is checked if it is 0 or 1. If $b_{k-1} = 1$ from the start pair $(P, 2P)$ we compute $(3P, 4P)$, otherwise $(2P, 3P)$. Continuing in this manner for $j = k - 2, \dots, 1$ we compute

$$\text{if } \; b_j = 1 : \; (L, R) \to (L + R, 2R),$$

$$\text{if } \; b_j = 0 : \; (L, R) \to (2L, L + R).$$

In our implementation this procedure is slightly changed (see section III).

The Second Phase

In the second phase of the algorithm for all primes $B_1 < p \leq B_2$, points pQ are computed. For each point pQ the program checks if the coordinates X_{pQ} and Z_{pQ} have a nontrivial common divisor with N. The constants B_1 and B_2 are chosen according to the size of the expected prime divisors of N. They are defined in the beginning of the program as input constants or automatically by the software. A variant of realization of the second phase procedure is to transform the curve into short Weierstrass form. However in Weierstrass form, the point calculation process requires the use of the extended Euclid algorithm (to invert a number modulo N). This slow up the performance of the algorithm, but increases the probability of finding a divisor of N.

3 Implementation Details and Computational Experiments

3.1 Description of the Implementation and Experiments

Our software implementation enables computations with Montgomery curves parameterized in two manner:

– "Standard" parametrization: We start with an initial point $P = (c : : 1)$, where c is an input value or it is chosen randomly by the software. Starting with a given d_1 (input parameter or randomly chosen) we compute d_2, d_3, d_4 by adding $4^{-1} \pmod N$. Then enlarge the set $\{d_1, d_2, d_3, d_4\}$ by adding 1 to each element until we obtain the numbers of parameters d equals to the number of threads.

– Suyama's parametrization: For each thread we generate a random σ and precompute the parameters $u,\ v,\ A$, and the initial point $P = (u^3 : : v^3)$.

We assign different value of A (indeed of d) for each thread. This assignment is realized in the described above manner.

In phase two we implement the classical algorithm (as it is described at the end of Section II) as well as a variant that consists of only doubling Q (or several pQ). In the latter case we collect the product of Z coordinates in a variable PROD_Z and periodically check its greatest common divisor with N.

The implementation of the ECM algorithm described above started from a new branch of the GMP-ECM project, which targets GPU devices [15]. It utilizes some of GMP's routines for the preprocessing/postprocessing steps, but transfers the greater deal of the computations on CUDA-enabled GPU devices. By further enhancing this branch with Message Passing Interface (MPI) and implementing the master-slave communication model we enable it to run on multiple GPUs. However, our implementation of this classical model differs greatly from the standard ones - instead of charging the master with both a part of the computations and tasks like preprocessing, postprocessing and distributing, we have chosen to completely separate the master from the main computations. In our implementation we allocate a single GPU for each slave process, while the master process has no GPUs. The master is charged only with small amount of preprocessing and postprocessing, and with the overall control of the slaves. Using this approach we separate the master process from most of the computations related to the actual factorization.

The master is charged with the task of partitioning and distributing the computations among the slave processes. During initialization the master collects information about the slave processes, mainly regarding the number of curves the GPUs can compute simultaneously. By doing this, the master can efficiently partition the curves in appropriate batches for different GPUs instead of using a fixed-size batch. Such an approach proves valuable when working with GPUs with different number of computing cores.

Another difference from the classical master-slave implementation is that in our implementation the slaves work asynchronously. In a single communication exchange the slave sends the results obtained from a batch of curves, while the master sends a new one using non-blocking MPI calls, thus speeding up the postprocessing of the results. Through the use of the MPI_ANY_SOURCE macro the master can receive the results from an arbitrary slave, which means that there is no explicit synchronization among the slave processes. This enables the usage of GPU devices with different clock-speed.

Separating the master from the main part of the computations, while granting it improved control over distributing of computations through the knowledge of each slave's GPU computational capabilities, along with enabling the slaves to work in asynchronous manner, allows for better utilization of the GPUs at the expense of CPU utilization. Since the greater part of the computations are done on the GPU devices this effect is desirable and increases number of processed curves.

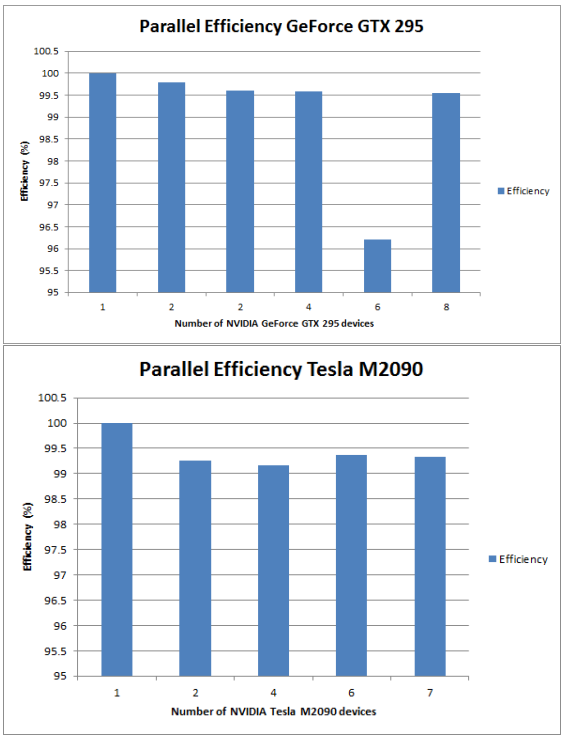


Fig. 1. Parallel efficiency

Introducing MPI bears some computational overhead. While in the original GPU implementation of GMP-ECM communication is restricted only between the CPU and GPU, in our implementation the computational tasks must be communicated twice - between the master CPU and slave CPUs and between each slave's CPU and it's GPU. Additionally, the usage of multiple GPUs on a single host demands that each process must specify the device explicitly in order to avoid mapping the same GPU device to different slave processes. This overhead varies between different GPUs and it is not considered in the presented numeric results. The asynchronous manner in which the slaves work leads to additional overhead, during the first communication exchange. However when using a great number of curves this overhead becomes negligible.

Since MPI and CUDA are "orthogonal", we decided to completely separate all GPU-related code and MPI code. This approach enabled us to employ different fine-tuning strategies on the different parts of the code and also allowed us to try different C compilers along with different MPI implementations for the MPI-related code.

3.2 Computational Resources Used

In our experiments we used two different NVIDIA GPU models - NVIDIA GeForce GTX 295 and NVIDIA Tesla M2090. The first one is a pure high-end gaming card, while the latter is a highly superior GPU specifically targeting GPGPU computations. In terms of raw processing power the Tesla M2090 offers 1331.2 GFLOPS in single precision and 665.6 GFLOPS in double precision for a single to double precision ratio of 2/1. On the other hand the inferior GeForce GTX 295 has a single to double precision ratio of 8/1 since its prime target is computer graphics where most of the computations are in single precision. However, it can deliver 1788.48 GFLOPS in single precision, which makes it a perfect platform for our ECM implementation, as the algorithm uses only integer arithmetic. It is also worth nothing that the GeForce GTX 295 supports only the limited compute capability 1.3, while the Tesla M2090 supports the enhanced compute capability 2.0.

Our numerical experiments were performed on two GPU clusters, both part of the HP-SEE infrastructure. The first one is based on NVIDIA GeForce GTX 295 and contains four nodes each equipped with a single GPU, which is a dual card and is visible to the operating system as two separate CUDA-enabled devices with 240 computing cores each. Each node runs CentOS 5.5 on Intel Core i7-920 processor at 2.66 GHz and 12 GB RAM. The nodes are interconnected via Gigabit Ethernet which is a limitation and requires a well-chosen communication pattern.

The second cluster contains two HP ProLiant SL390s G7 servers, each equipped with two Intel Xeon E5649 processors and 96 GB RAM. Each server runs Scientific Linux 5.4. The servers are connected via DDR InfiniBand. Each of the servers can be equipped with a maximum of eight NVIDIA Tesla M2090. Currently one of the servers is equipped with six GPUs and the other is equipped with one, which allows us to effectively measure the efficiency of our MPI communication pattern.

3.3 Obtained Results

Table 1 and the two figures represent the results of our numerical experiments performed on the two clusters. The best results on both GPU devices were obtained when using the maximum possible number of threads per block, which is 512 and 1024 for the NVIDIA GeForce GTX 295 and NVIDIA Tesla M2090, respectively. Usage of smaller block sizes proved less efficient due to poorer utilization of the available shared memory. Additionally we compared two C compilers - GCC 4.1.2 and Intel C compiler 12.1.0, as well as two MPI implementations - GCC-compiled OpenMPI and Intel MPI Library 4.0. The presented results were obtained using the Intel C compiler along with Intel MPI Library, as this resulted in a slight improvement in comparison to the other combinations of compiler and MPI implementation.

The parallel efficiency is evaluated, by comparing the performance of the original non-MPI application to our MPI-enabled implementation. The column

Table 1. Numerical results

GeForce GTX 295

Execution scheme	Running time	Peak performance (GFLOPS)	Curves per second	UF
1 node with 1 device	1730.32	894.24	355.08	0.39707
1 node with 2 devices	866.98	1788.48	708.67	0.39624
2 nodes with 1 device each	868.61	1788.48	707.33	0.39549
2 nodes with 2 devices each	434.43	3576.96	1414.28	0.39539
3 nodes with 2 devices each	299.74	5365.44	2049.76	0.38203
4 nodes with 2 devices each	217.29	7153.92	2827.52	0.39524

Tesla M2090

Execution scheme	Running time	Peak performance (GFLOPS)	Curves per second	UF
1 node with 1 device	1257.30	1331.20	488.67	0.36709
1 node with 2 devices	633.38	2662.40	970.03	0.36435
1 node with 4 devices	316.94	5324.80	1938.52	0.36405
1 node with 6 devices	210.90	7987.20	2913.26	0.36474
1 node with 6 devices and 1 node with 1 device	180.81	9318.40	3398.00	0.36466

Stage 1 bound: 300; Number of Curves: 614400;
Number N: 266957047625186069027747382703973.
UF - utilization factor = Curves per second / Peak performance.

"Peak performance" in Table 1 represents the sum of the peak performance of the GPU devices used in the corresponding run, where the peak performance of each device is as per its hardware specifications. Our implementation showed excellent results, with lowest efficiency of over 96% on the cluster with Gigabit Ethernet interconnect, and over 99% on the cluster connected via DDR InfiniBand.

4 Conclusion and Future Work

Our parallel implementation of GMP-ECM demonstrated excellent parallel efficiency on both of the GPU clusters. Naturally, this scalability can only be achieved when using a large number of curves, substantially larger than the number of computing threads. The chosen implementation of the master-slave model handles well GPUs with both different clock-speed and different number of CUDA cores and requires no effort for porting on different GPUs. Applications with mostly integer arithmetic run efficiently on both General-purpose computing GPUs (GPGPUs) and gaming GPUs, which turn the latter into a suitable and inexpensive platform for our implementation of the ECM. In some situations, it may not be possible to interconnect all GPU resource with compatible versions of MPI. This is a motivation to explore alternative lower-level methods of communication over protocols like UDP.

Acknowledgments. This work was partially supported by the European Commission under EU FP7 project HP-SEE under contract number 261499 and by the Bulgarian National Science Fund under Contracts DCVP02/1 CoE Super CA++.

References

1. Bernstein, D., Lang, T.: Explicite formulas database, http://hyperelliptic.org/EFD
2. Bernstein, D.J., Birkner, P., Lange, T., Peters, C.: ECM using Edwards curves (2008), http://eprint.iacr.org/2008/016
3. Bernstein, D.J., Lange, T.: Faster addition and doubling on elliptic curves. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 29–50. Springer, Heidelberg (2007)
4. Bernstein, D.J., Birkner, P., Joye, M., Lange, T., Peters, C.: Twisted Edwards curves. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 389–405. Springer, Heidelberg (2008)
5. Bernstein, D.J., Chen, T.-R., Cheng, C.-M., Lange, T., Yang, B.-Y.: ECM on Graphics Cards, http://eprint.iacr.org/2008/480
6. Bernstein, D.J., Chen, H.-C., Cheng, C.-M., Lange, T., Niederhagen, R., Schwabe, P., Yang, B.-Y.: ECC2K-130 on NVIDIA GPUs, http://eprint.iacr.org/2012/002
7. CUDA 4 Programming Guide, http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf
8. Dongarra, J., et al. (eds.): Sourcebook of Parallel Computing. The Morgan Kaufmann Series in Computer Architecture and Design. Morgan Kaufmann Publisher, Elsevier Science (2003)
9. Edwards, H.M.: A normal form for elliptic curves. Bulletin of the American Mathematical Society 44, 393–422 (2007), http://www.ams.org/bull/2007-44-03/S0273-0979-07-01153-6/home.html
10. Lenstra Jr., H.W.: Factoring integers with elliptic curves. Annals of Mathematics 126, 649–673 (1987)
11. Montgomery, P.L.: Speeding the Polard and elliptic curve methods of factorization. Mathematics of Computation 48, 243–264 (1987)
12. MPI-2 Standard, http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf
13. Szerwinski, R., Güneysu, T.: Exploiting the Power of GPUs for Asymmetric Cryptography. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 79–99. Springer, Heidelberg (2008)
14. Zimmermann, P., Dodson, B.: 20 Years of ECM. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 525–542. Springer, Heidelberg (2006)
15. Zimmermann, P., et al., https://scm.gforge.inria.fr/svn/ecm/trunk/gpu/

Advanced Vulnerability Assessment Tool for Distributed Systems

Sandor Acs[1,2], Miklos Kozlovszky[12], and Peter Kotcauer[1,2]

[1] University of Obuda, H-1300 Budapest, P.O. BOX 267, Hungary
acs.sandor@biotech.uni-obuda.hu
[2] MTA SZTAKI Computer and Automation Research Institute, H-1518 Budapest, P.O. Box 63, Hungary

Abstract. Large-scale high performance systems have significant amount of processing power. One example of such system is the HP-SEE's HPC and supercomputing infrastructures, which is geologically distributed, and provides 24/7, high performance/high throughput computing services primarily for high-end research communities. Due to the direct impact on research and indirectly on economy such systems can be categorized as critical infrastructure. System features (like non-stop availability, geographically distributed and community based usage) make such infrastructure vulnerable and valuable targets of malicious attacks. In order to decrease the threat, we designed the Advanced Vulnerability Assessment Tool (AVAT) suitable for HPC/supercomputing systems. Our developed solution can submit vulnerability assessment jobs into the HP-SEE infrastructure and run vulnerability assessment on the infrastructure components. It collects assessment information by the decentralized Security Monitor and archives the results received from the components and visualize them via a web interface for the local/regional administrators. In this paper we present our Advanced Vulnerability Assessment Tool, we describe its functionalities and provide its monitoring test results captured in real systems.

Keywords: distributed computing, HPC, security, vulnerability assessment, HP-SEE, supercomputing infrastructure.

1 Introduction

1.1 Large Scientific Computing Resources

Core European e-Infrastructure for large-scale e-Science research consists of distributed computing infrastructure (DCI), distributed storage infrastructure (DSI) and backbone (e.g GANT) network. Large number of initiatives and projects builds up the infrastructure. The SEE-GRID [1] initiative with three consecutive projects (SEE-GRID, SEE-GRID2 and SEE-GRID-SCI) provided the local grid based DCI in the SEE (South-East European) region. The HP-SEE project [2] links together existing and upcoming HPC facilities in the region in a common infrastructure to open up the HPC infrastructures to a wide range

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_17, © Springer International Publishing Switzerland 2014

of new user communities, including those of less-resourced countries facilitating cross-border research and collaboration.

1.2 Motivation

There are lots of potential security problems with distributed and shared systems caused by the technology itself and by site/software stack setup or bring on by end-user behaviour.

Our general goal is to provide assistance for system administrators building up sustainable and less vulnerable infrastructure and survive cyber-attacks. For this aim we have created a vulnerability assessment framework for distributed systems in order to decrease threats.

In this paper in Section II we briefly introduce recent vulnerability trends. In Section III we provide information about existing security monitoring solutions. In Section IV we give detailed description about the design and implementation of the developed Advanced Vulnerability Assessment Tool. Later on in Section V we show vulnerability test results of AVAT. At the end of our paper we conclude and explain directions of our future work.

1.3 Vulnerability Trends

The software evolution (size, functionality set, etc.) generally increases complexity in the software stack. It is hard to protect even a single PC node infrastructure against malicious attacks. This problem multiplies significantly if the infrastructure is heavily distributed, contains thousands of cores, and serves hundreds of people. Security issues are constantly explored worldwide and published on the relevant on-line sites as Common Vulnerabilities and Exposures (CVE) [3]. Common Vulnerability Scoring System (CVSS) Initiative [4] was funded by the U.S. Department of Homeland Security in order to provide vendor independent and common scoring system of known vulnerabilities. The scores are integer numbers between 0 and 10.

We analysed the CVSS databases and we can point out that the highest threat was from 2006 to 2008 in the last 10 years. Figure 1 shows the number of published vulnerabilities and the total severity, which was calculated from the number of vulnerabilities multiplied by their CVSS score.

Figure 2 presents that the average of CVSS score is about 6 and their standard deviation is about 2 in every year in the investigated period. One of the biggest problem is, that the successful break-in method (which are definitely very hard to find cyber-attack solutions) against any of the DCI infrastructure elements is very likely reusable and provide vast gain for the intruder.

2 Related Work

The SEE-GRID projects (SEE-GRID-SCI project and its predecessors) are using two tools and three services to monitor their grid infrastructure: Hierarchical

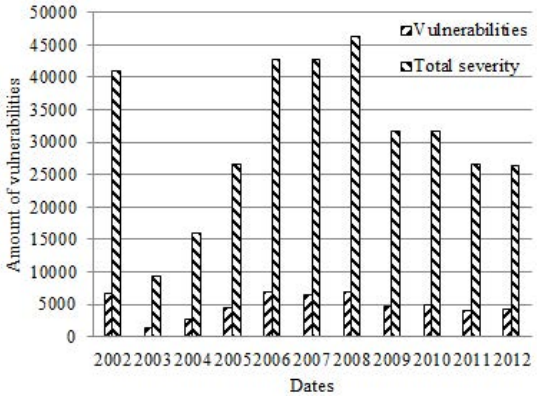


Fig. 1. Trends in number of published vulnerabilities and their total severity

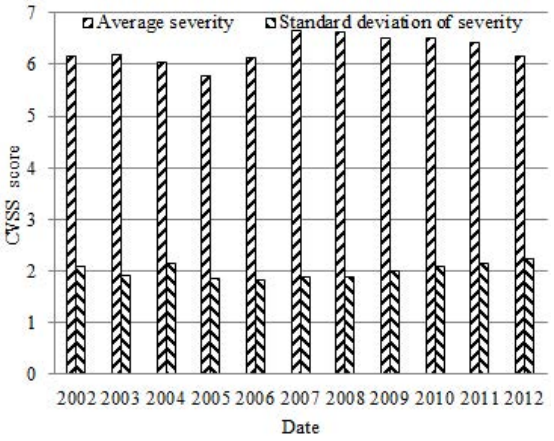


Fig. 2. Average and standard deviation of severity

Grid Site Management (HGSM) [5] that is the central information tool. HGSM is a web based database designed to hold static information about the grid sites. This information bundle includes physical location of the site, emergency contact information, operating system and middleware of the grid site, detailed hardware information of nodes and site downtimes. BBmSAM [5] portal that submits regular SAM (service availability monitoring) tests every 3 hours for interactive tests and every hour for non-interactive tests. The BBmSAM also provides data on site and service availability for the needs of the project. The PAKITI [6] service, which was used for security status monitoring of the infrastructure, collects the list of the installed software packages on the nodes and matches the gathered information with the security database (coming from an external repository). It is using HTTP or HTTPS protocol to communicate and provide a graphical user interface for its users. However the program has major serious

disadvantages in distributed IT environment (e.g grid administrators have to install the client software on every machine and and they should configure the firewall settings as well).

The Grid Site Software Vulnerability Analyzer (GSSVA) [7] tool aimed to ease the problems of PAKITI. Therefore, the GSSVA uses only essential grid protocols in order to work on every site without changes in configuration or installing any software. However, a vulnerability analyzing should have more kind of security audit procedures than only comparing software versions (that PAKITI and GSSVA does). There are some widely used closed and open-source security analyzer frameworks that can provide various types of security information about the investigated infrastructures.

Qualys Guard [8] offers a complete vulnerability management solution. The centralized front-end component is hosted by Qualys, thus the users only need a web browser to manage the target assets, schedule the vulnerability scans, generating customized reports and deal with the remediation tracking. As it is a cloud based service, the attacks are coming from the internet.

The OpenVAS (Open Vulnerability Assessment System) [9] is an open source vulnerability assessment software. It consist about three layers of components. The first layer is the user interface, where users or automatized scripts can access the system. The second layer is the services layer and the third layer is the data layer which means configurations, scanning templates and results. OpenVAS uses Network Vulnerability Test (NVT) for checking the IT resources.

The presented tools are complex and feature-rich, however they cannot be used easily in distributed systems because the HPC centers or grid sites are located on different administration domains. Therefore, the centralized management of these tools could not be satisfied.

3 Design and Implementation

We have followed three basic rules for designing the AVAT. (1) Only open-source software components shall be used to build up the system. (2) The solution should work in the users space of the targeted DCI. We shall not suppose more opportunity to reach the resources (e.g open ports on firewalls, use administrator privileges) than a generic user has, because the investigated infrastructures are located in different administration domains. Therefore, we used only DCI specific protocols and middleware services to deploy services, query/retrieve information or communicate with the DCI resource (e.g job submission). (3) Critical infrastructure assessment should always be carried out with the highest caution. Harmful tests shall not be used during the investigations on the live system to avoid service interruption.

3.1 Architecture

Figure 3 presents the high level schematic system plan with all the AVAT modules. The control module is responsible for connecting the different modules and

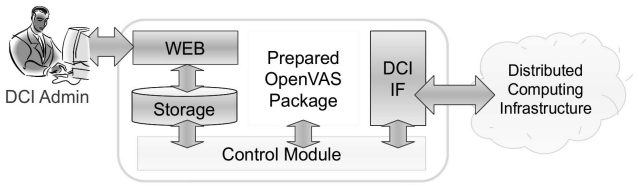


Fig. 3. AVAT modules

scheduling the vulnerability scans. The tool can be used with ARC [10] and gLite [11] based DCI interface module (majority of the HP-SEE infrastructure is using gLite and ARC, as EGIs grid infrastructure does it in a similar way). These modules contain the middleware specific commands to copy and run the vulnerability scanner and to gather the results of the investigations. The prepared OpenVAS package contains a modified and precompiled OpenVAS vulnerability scanner. The AVAT stores the scan results and makes them available for the administrators of the resources.

3.2 Introducing gLite Based Resources

The following sub-section introduces a brief overview about some components of the gLite middleware.

On the User interface (UI), a user can gain access (after the successful authentication) to use the DCI resources. The UI provides command line interface to the end user to utilize all the basic Grid operations. Computing Element (CE) is some set of computing resources localized at a site (i.e. a cluster, a computing farm). Worker Nodes (WN) are execute the tasks. A Storage Element (SE) provides uniform access to data storage resources. The Information Service (IS) provides information about the grid resources and their status. Berkeley Database Information Index (BDII) is used to store and publish monitoring and accounting data from the grid sites. Virtual Organization Membership Service (VOMS) is tightly coupled system for managing authorization data within multi-institutional collaborations.

So far we know how a gLite based middleware builds up, we can describes how the AVAT checks the targeted grid resources. We need to note here, that AVAT is a generic solution and capable to assess both gLite and ARC based DCIs. Investigation of ARC based resources is very similar, only the component names and the used commands are different. The AVAT uses the following steps in order to check gLite base resources: The control module sets up the environment (e.g name of the investigated VO and the entry point of the information system). Then, the control module collects the available resources from the information system and it sends test jobs into the resources individually via the DCI module.

Figure 4 shows the object interactions when AVAT investigates a gLite based infrastructure. The DCI module uses a gLite UI to submit the test jobs to the WMS. The WMS forward the job to the corresponding CE. The CE schedules

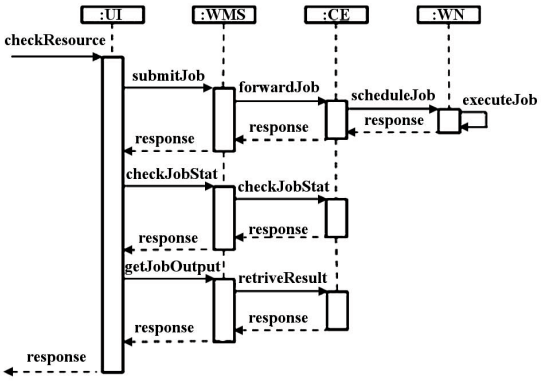


Fig. 4. Investigating a gLite resource by AVAT

the task to one of its WNs. The WN execute the test job and the CE sends back the results to the WMS. The AVAT checks the status of the job periodically and if it is ready than the results will be gathered by the UI. The test job (that runs on WNs) contains the following sub- tasks: (1) Download the precompiled OpenVAS server, client and libraries. (2) Set up the environment (e.g set up the path of the binaries and the libraries). (3) Update the collection of the NVTs. (4) Start the OpenVAS server. (5) Start the OpenVAS client, connect to the server and scan the local machine.

4 Investigating the Results of the Test Scans

To prove AVATs capabilities and test its usability parameters, both gLite and ARC based DCIs have been investigated during autumn 2012.

4.1 SEE-GRID VO

The AVAT queries the available resources from the info-system of the SEE-GRID gLite based grid VO and sends test jobs into the corresponding sites. These test jobs characteristically run from 20 to 30 minutes. There is a parameter in the system that determines the delay of collecting the results. This parameter is one hour by default, however it can be freely adjusted to the underlying DCI.

Figure 5 summarizes the results of the investigation. Most of the sites have vulnerability issues. There are two sites having more than 10 security holes and the average is 3.5/site. Only 4 from the 16 investigated sites were up-to-date or secured with hardened kernel.

4.2 HP-SEE Supercomputing Infrastructure

The HP-SEE supercomputing infrastructure does not have any centralized authorization entry point (like the VOMS in the gLite based grids). Therefore, we

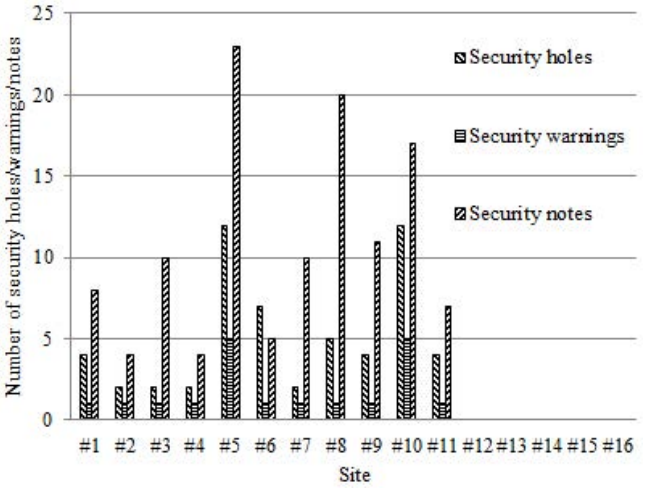


Fig. 5. Vulnerability status of the SEE-GRID VO

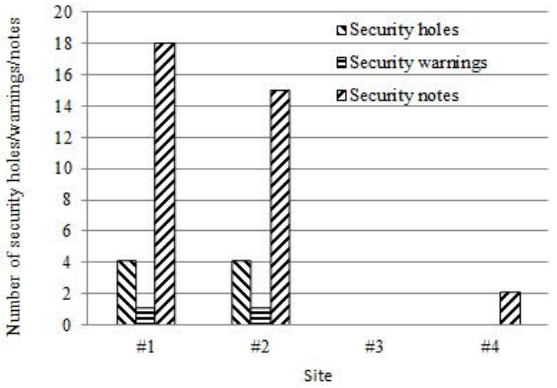


Fig. 6. Vulnerability status of the investigated HPC centers

registered into four sites of HP-SEE. The AVAT test jobs had been sent to the HP centers separately and the results were collected about 5 minutes later.

Figure 6 summarizes the results of the investigation. The integrated OpenVAS service reported four security holes and numerous (15 and 18) security warnings for site 1 and site 2. There were not detected any vulnerability on the third and fourth sites.

5 Conclusion and Future Work

In this paper, we discussed the major vulnerability sources in distributed computing infrastructures and we presented some of the recently used monitoring

solutions and vulnerability frameworks in DCIs (highlighted the SEE-GRID-SCI and HP-SEE DCIs). We pointed out that the centralized vulnerability assessment solutions for distributed systems (PAKITI and GSSVA) only do software version checks. However, the feature-rich frameworks (e.g OpenVAS) could not work easily in different administrations domains. Then, we introduced our proposed software solution; the Advanced Vulnerability Assessment Tool (AVAT) that combines the benefits of the GSSVA and OpenVAS frameworks. Moreover, we discussed the results of using AVAT on two different type of DCI: grid SEE-GRID-SCI and supercomputing infrastructure HP-SEE, based on two different middleware (gLites and ARC).

In the future, we plan to extend the AVAT with more DCI interface modules: we would like to add EC2 interface in order to support cloud vulnerability assessment too. The proposed software can easily (re)used for other DCIs and authors are jointly working with various DCI communities to open up the vulnerability service for other projects and infrastructure providers.

Acknowledgment. The research leading to these results has received funding from the European Social Fund and the Hungarian TMOP-4.2.1.B-11/2/KMR-2011-0001 Kritikus infrastruktura vedelmi kutatasok project and as well as from European Union Seventh Framework Programme (FP7/2008-2013) under grant agreement no RI-261499 (HP-SEE) High-Performance Computing Infrastructure for South East Europes Research Communities project.

References

1. The SEE-GRID-SCI website (2012), `http://www.see-grid-sci.eu/`
2. Kozlovszky, M., Windisch, G., Balasko, A.: Short fragment sequence alignment on the HP-SEE infrastructure. In: MIPRO, 2012 Proceedings of the 35th International Convention, May 21-25, pp. 442–445 (2012)
3. Martin, R.A.: Managing Vulnerabilities in Networked Systems. IEEE Computer Society Computer Magazine, 32–38 (2001), `http://cve.mitre.org/`
4. Mell, P., Scarfone, K., Romanosky, S.: A complete guide to the common vulnerability scoring system, version 2.0. Forum of Incident Response and Security Teams (June 2007)
5. Balaz, A., Prnjat, O., Vudragovic, D., Slavniv, V., Liabotis, I., Atanassov, E., Jakimovski, B., Savic, M.: Development of Grid e-Infrastructure in South-Eastern Europe. J. Grid. Comput. (9), 135–154 (2011)
6. The Pakiti website (2012), `http://pakiti.sourceforge.net/`
7. Acs, S., Kozlovszky, M., Balaton, Z.: Automation of security analysis for service grid systems. In: PARENG 2009, The First International Conference on Parallel, Distributed and Grid Computing for Engineering, Pcs, Hungary (2009)
8. The Qualys website (2012), `http://www.qualys.com/`
9. The OpenVAS website (2012), `http://www.openvas.org`
10. The ARC website (2012), `http://www.nordugrid.org/arc/`
11. The gLite website (2012), `http://glite.web.cern.ch/glite/`

Performance and Scalability Evaluation of Short Fragment Sequence Alignment Applications

Gergely Windisch[1], Miklos Kozlovszky[1], and Ákos Balaskó[2]

[1] Óbuda University/John von Neumann Faculty of Informatics, Budapest, Hungary
[2] MTA SZTAKI/Laboratory of Parallel and Distributed Computing, Budapest, Hungary
windisch.gergely@nik.uni-obuda.hu

Abstract. Mapping short fragments to open access eukaryotic genomes at a very large scale presents a data processing challenge to the scientific world. The main tool used for such an application is BLAST which is the one we use in our portlets developed at Obuda University.

Lately most of our work has been focused on evaluating the performance and scalability of our applications by profiling, analyzing the results of the tests and improving the performance of both the portlets and the server-side massively parallel algorithm by environment optimization using the data collected during the testing phase.

In this paper we will describe the two portlets (Deep Aligner and Disease Gene Mapper), discuss the issues and challenges during the development and the performance analysis and present our results on the performance and scalability of the applications.

Keywords: Application porting, sequence alignment workflow, HP-SEE, gUSE, scalability, mpiBlast.

1 Introduction

An important research topic nowadays in the field of bioinformatics is short segment alignment. Mapping the short sequences to large DNA databases is a very computation intensive process because of the large size of the DNA databases and the large number of alignments required for a query. This data- and computation intensive algorithm results in extremely long runtimes, so a solution is needed to decrease the computational time. One way for the scientists to solve said problem is to use eScience Gateways for problems in bioinformatics like the MOSGrid portal [1,2] or ProGenGrid [3]. The High-Performance Computing Infrastructure for South East Europe (HPSEE) project [4] connects existing and upcoming HPC computing centers into a common infrastructure and provides high performance computing to researchers in the area. As part of that project, we at Obuda University and MTA SZTAKI have implemented and manage a portal called Bioinformatics eScience Gateway which is a gUSE/WS-PGRADE [5] framework for providing services for the

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_18, © Springer International Publishing Switzerland 2014

scientists in forms of portlets running on the server. On the gateway the gUSE back-end is connected to multiple DCIs. The implementation of the framework make the WS-PGRADE workflows that the portlets use independent of the DCI middleware which results in th the system being highly scalable and more resources can be added later without having to modify the existing services.

For more information on the Bioinformatics eScience Gateway please see [6]. In this paper we will describe the two portlets (Deep Aligner and Disease Gene Mapper) that we have developed and provide on the Bioinformatics eScience Gateway.

2 HP-SEE's Bioinformatics eScience Gateway

The Bioinformatics eScience Gateway is based on gUSE [7] and operates within the Life Science VO of the HP-SEE infrastructure. It provides a unified interface for different bioinformatics services like BLAST and gene mapper applications. The graphical front-end of gUSE is called WS-PGRADE. gUSE is implemented as web services and all of its services can be accessed using a simple web browser. It currently holds two portlets which we will discuss later in the chapter, but it is an open framework that accepts gUSE ASM based applications created by fellow researchers.

The two portlets use the Application Specific Modules (ASM) API of the gUSE [8]. Using the ASM API the users can create a user friendly graphical front-end that invokes the WS-PGRADE workflows in gUSE.

2.1 Disease Gene Mapper

The Disease Gene Mapper service allows researchers to utilize the HPC infrastructure to find gene sequences in an organism which have already been identified as a disease in a different organism. The users of DGM have to provide the "source" disease name and an organism, and a second organism against which the gene sequence search will be executed. The portlet then connects to the NCBI website and downloads all the latest results (sequences) linked to the given disease. The resulting sequences will then be searched in the selected database of the given organism.

2.2 Deep Aligner

Deep Aligner portlet (Figure 1) allows researchers to search for a multitude of short gene sequences in a given organism. The users can upload multiple sequences in a compressed file (.rar, .zip or .tar.gz), the portlet searches for all of them in the selected database. For both portlets the user can set the parameters for the BLAST search like the e-value. Parameters like the number of MPI cores and other relevant settings concerning the HPC infrastructure have thoroughly been researched and fine tuned (see Chapters 3 and 4 in this paper) and cannot be changed by the end users.

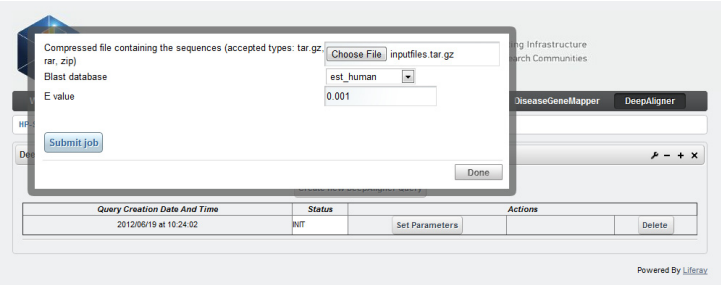


Fig. 1. Deep Aligner new query screen

3 Performance Analysis

The two applications we have created share some of the code base which results in a similar behavior. Both applications consist of three jobs in a WS-PGrade workflow with job 1 being the preprocessor, job 2 doing the calculations and job 3 collecting the results and providing them to the user.

3.1 Job 1

The difference between the two applications are the largest at Job 1. In the case of Deep Aligner, Job 1 gets a tar.gz containing all the input sequences the user specifies. It decompresses them and starts multiple instances of Job 2. The current implementation for the decompression is serial, we have investigated using a parallel decompressor but according to our profiling approximately 0.01 % of the total execution time is spent on Job 1 in Deep Aligner, so parallelizing it would yield no real performance gain.

Job 1 in Disease Gene Mapper is more complex – the main input for the job is the name of a disease (e.g. Asthma) which it looks up on the appropriate NCBI website and downloads the DNA sequences associated with the given disease. This part is done using a simple wget which is also serial. It would be possible to do parallel downloading but according to our testing the NCBI website does not allow too many concurrent downloads from the same computer.

Our conclusion is that Job 1 is not an ideal candidate for parallelization because the execution timeframe is too short and remote services don't allow parallel access.

3.2 Job 2

Job 2 is the "workhorse" for both our portlets. It is essentially the same for both applications. The main algorithm is BLAST which implemented by calling mpiBlast. Or profiling shows that 99.4% of the total execution time is spent on job 2 – mainly

mpiBlast. Further investigation reveals [9] that 85% of that time is spent on the actual BLAST search, 7% is on fragment copy and communication and 3% is on printing the results. Other functions use up the rest of the time.

3.3 Job 3

Job 3 collects the results from all the mpiBlast executions, creates a compressed file and sends it back to the gUSE portlet. It too is a sequential job, but it takes up so little of the total execution time that it is better to leave it as it is.

We can conclude then, that job 2 is the main candidate for performance optimization. We have chosen mpiBlast because of its proven speed and reliability. The measurements described in the following sections therefore only include the results from mpiBlast and were executed on one of the HPC centers operated by NIIF called "Budapest", which is an HP cluster consisting of 32 nodes with 24 CPU cores each (i.e. total number of CPU cores: 768). The mpiBlast executable was compiled using the openMPI-gcc. The BLAST database size was 5.1 GB, and the input sequence size was 29.13 kB. Each measurement was executed 10 times, the average of the 10 executions was taken as the final result.

4 Performance Measurement Results

4.1 Runtime Benchmark

The following graphs show the result of multiple executions of mpiBlast on the same database with the same input sequence on the same computer. The only difference being the number of CPU cores allocated to the MPI job.

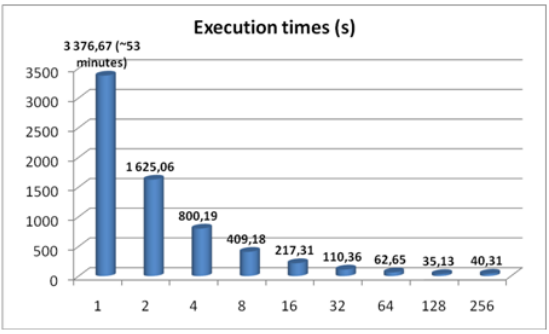


Fig. 2. Execution times measured by mpiBlast in seconds

Figure 2 shows the execution times measured by mpiBlast. If executed on just one CPU it takes 3376 seconds for the job to finish (about 53 minutes). As we can see the applications scales well, the execution times drop when we add more and more CPUs.

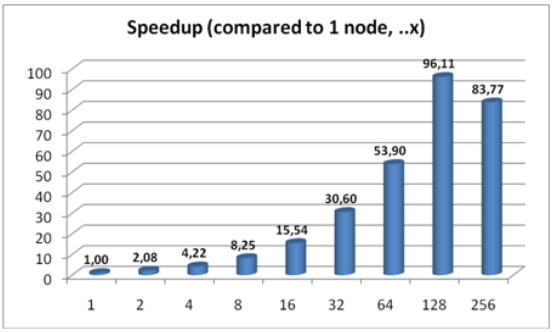


Fig. 3. mpiBlast speedup using multiple CPU cores compared to running it on just one CPU core

Figure 3 shows the speedup in percentage compared to the original measurement on one CPU core. The results show that the application loses momentum at around 32 cores but the performance increases until around 128 cores. Figure 4 shows the same results but from a different angle: that of the efficiency – the speedup / number of cores.

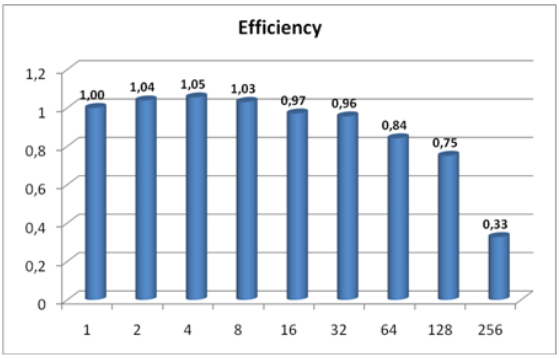


Fig. 4. Efficiency of using multiple CPU cores

Ideally in a perfectly scaling application the numbers should be around one. As we can see from the graph the efficiency is quite high (>75%) until the number of cores reaches 128 where it starts to drop.

4.2 DB Fragment Numbers

The first task when using mpiBlast is to split the BLAST database into multiple fragments. According to [10] the number of database fragments has a direct impact on the performance of the application. Finding an optimal number was essential, so our database was split into different sizes. Figure 5 shows the measured execution times. The measurements were executed on 64 cores.

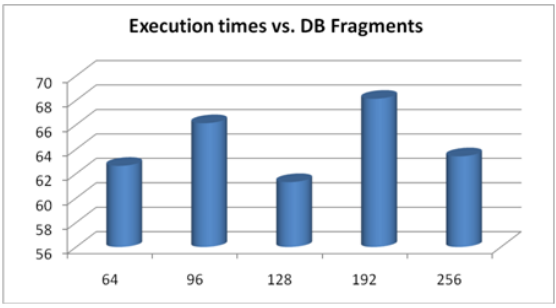


Fig. 5. Execution times in seconds using different number of Database Fragments

As it is apparent from the graph, the application performs best when the number of DB segments are integer multiples of the number of CPU cores. This result coincides with the results published in [10]. The reason is straightforward: this is the only way an even data distribution can be achieved amongst the cores.

4.3 Real World Performance

Based on Figure 3 we could conclude that the optimal number of CPU cores for this workload and computer is 128 where the efficiency is still quite high. However, we have found that there is another factor that has to be taken into account, and that is wall clock time. Due to inner workings of the SGI environment jobs are scheduled for execution when there is enough free CPU cores to satisfy the amount required by the given job and smaller jobs (i.e. those requiring less CPU cores) are favored to those requiring lots of them. These two factors together result in large jobs spending a lot of time on the queues – at the end of the day jobs using more CPUs actually finish later. The following figures show results calculated from the wall clock time (the actual time it took from pressing the start button until the results came back).

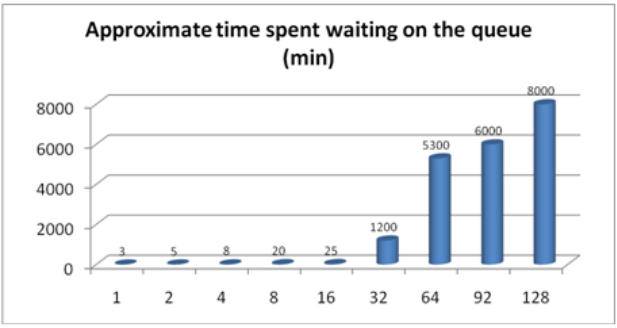


Fig. 6. Average time in minutes mpiBlast jobs had to wait on the queue before being executed

Figure 6 shows the approximate time spent waiting on the queue (as opposed to the previous graphs, the values on the Y axis are minutes). As we can see the waiting time is not too bad until 16 cores, it takes less than half an hour for the job to be scheduled. The numbers go up at 32 and 64 cores, and on average a 128 core job has to wait more than five days.

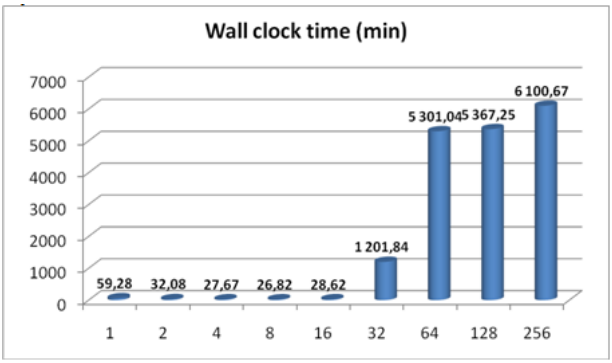


Fig. 7. Wall clock time in minutes – the waiting time in the queue plus the execution time

Figure 7 shows the wall clock time – waiting + execution time[1]. The results are similar to the previous graph, until 16 cores the speedup of the execution makes up for the longer queue, but it gets out of hand at 32 cores.

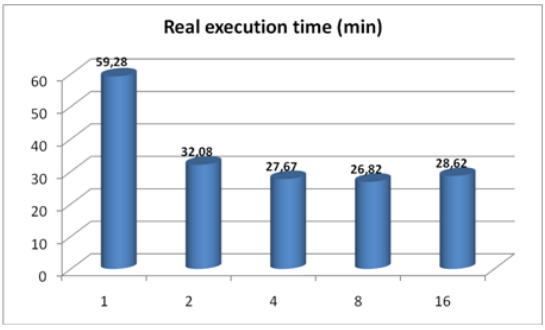


Fig. 8. Wall clock time showing the results for CPU cores less than 32

Figure 8 shows the same wall clock times as Figure 7 but core numbers above 16 are omitted for better readability. Cores between 4 and 16 perform quite similarly, considering all the factors we have chosen 16 cores for our application because at times of less utilization it spends less time on the queue and results in even faster execution.

[1] The results on Figure 7 come from a different measurement than Figure 6.

5 Conclusion

In this paper we have described the Deep Aligner and Disease Gene Mapper portlets developed at Obuda University and hosted on the HPSEE Bioinformatics eScience Gateway. The two portlets can be used to do short sequence alignments against large databases in a short time using the supercomputing infrastructure at NIIF, Hungary. We also describe the performance analysis and tuning done on the applications.

6 Future Work

There are still many things to be done with our two portlets for them to be even better assets to the scientific community. Our future plans are threefold:

- Add new features to the portlets
- Make them available on more HPC infrastructures
- Improve performance by further application tuning – different compilers, compiler options.

References

1. Gesing, S., et al.: Granular security for a science gateway in structural bioinformatics. In: Proc. IWSG-Life 2011 (2011)
2. Gesing, S., et al.: Special issue: Portals for life sciences|providing intuitive access to bioinformatic tools. Concurrency and Computation 23(3), 223–234 (2011)
3. Aloisio, G., Cafaro, M., Fiore, S., Mirto, M.: ProGenGrid: a workflow service infrastructure for composing and executing bioinformatics grid services. In: Proceedings. 18th IEEE Symposium on Computer-Based Medical Systems, June 23-24, pp. 555–560 (2005), doi:10.1109/CBMS.2005.90
4. HP-SEE core presentation, `http://www.hp-see.eu/files/HPSEE-WP1-GR-027-CorePresentation-a-2010-11-22.pdf`
5. Kacsuk, P., Karoczkai, K., et al.: WS-PGRADE: Supporting parameter sweep applications in workflows. In: Proc. of 3rd Workshop on Workflows in Support of Large-Scale Science, Austin, TX, USA, November 17, pp. 1–10 (2008) ISBN: 978-1-4244-2827-4
6. Kozlovszky, M., Windisch, G., Balasko, A.: Short fragment sequence alignment on the HP-SEE infrastructure. In: MIPRO, 2012 Proceedings of the 35th International Convention, May 21-25, pp. 442–445 (2012)
7. Kacsuk, P., Farkas, Z., Sipos, G., Hermann, G., Kiss, T.: Supporting Workflow-level PS Applications by the P-GRADE Grid portal. In: Towards Next Generation Grids Proceedings of the CoreGRID Symposium 2007 (2007)
8. Balaskó, Á., Kozlovszky, M., Schnautigel, A., Karóczkai, K., Márton, I., Strodl, T., Kacsuk, P.: Converting P-GRADE grid portal into e-science gateways. In: Proceedings of International Workshop on Science Gateways, IWSG 2010, Catania, pp. 1–6 (2010)
9. Lin, H., Balaji, P., Poole, R., Sosa, C., Ma, X., Feng, W.: Massively Parallel Genomic Sequence Search on the Blue Gene/P Architecture. In: IEEE/ACM SC 2008 (November 2008)
10. Darling, A., Carey, L., Feng, W.: The Design, Implementation, and Evaluation of mpiBLAST. In: 4th International Conference on Linux Clusters (June 2003)

Implementation and Benchmarking of New FFT Libraries in Quantum ESPRESSO

Dušan Stanković, Petar Jovanović, Aleksandar Jović, Vladimir Slavnić, Dušan Vudragović, and Antun Balaž

Scientific Computer Laboratory, Institute of Physics Belgrade, University of Belgrade, Pregrevica 118, 11080 Belgrade, Serbia
{dusan.stankovic,petar.jovanovic,aleksandar.jovic,vladimir.slavnic, dusan.vudragovic,antun.balaz}@ipb.ac.rs

Abstract. Quantum ESPRESSO (QE) software package allows electronic-structure calculations and materials modeling at the nanoscale, based on density-functional theory, plane waves, and pseudopotentials. It extensively uses Fast Fourier Transform (FFT) during all computations. In addition to the built-in FFT libraries, QE enables integration of newly developed FFT algorithms. Since Fastest Fourier Transform of the East (FFTE) library has shown performance comparable with the widely used and vendor-supplied libraries, the same behavior is foreseen in QE. In this paper we present FFTE-enabled and thread-enabled FFTW3 extensions of QE, together with benchmarking and performance results.

Keywords: FFT, Quantum ESPRESSO, multithreading, hybrid parallelism, OpenMP, MPI.

1 Introduction

Quantum Espresso is an integrated suite of open-source codes for electronic structure calculations and materials modeling at the nanoscale. It is based on density-functional theory, plane waves and pseudopotentials [1].

Fourier transformation is used in a large part in calculations performed in QE, so any gains in FFT performance would be positively reflected in the performance of the entire QE suite. Most major hardware platforms, along with their corresponding numerical libraries, are already supported in QE (such as IBM ESSL, Intel MKL, SGI SCSL and so on), which include routines for FFT calculations. Also, the open-source FFTW (version 2) and FFTW3 libraries [2] are supported.

Parallelization in Quantum ESPRESSO is achieved using MPI and OpenMP, and hybrid parallelism using both MPI and OpenMP together is currently supported only with the internally supplied FFTW library. The work on implementing the support for the open-source FFTE library was motivated by its performance results [3], so it was expected to show better performance than the open-source libraries already supported in QE. The work on implementing the support for hybrid FFTW3 library was considered because hybrid parallelism

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_19, © Springer International Publishing Switzerland 2014

is becoming more important, as computing nodes on modern HPC systems often comprise many CPU cores. Since the open-source FFTW3 library is widely used, and has both multi-threaded routines, and serial thread-safe routines, it was selected for implementation.

2 Quantum ESPRESSO Code Structure and Applied Modifications

Quantum ESPRESSO is written mostly in FORTRAN 90. It has a modular structure, with different modules for higher level domain specific calculations (for example, CP or PW modules), and also some general purpose parts which are then used in many other modules (for example FFT calculations or time logging).

The development of this project used QE 5.0 as a baseline, and was localized to the parts of the code responsible for FFT calculations. Analysis of the QE source code revealed that all the routines for performing FFT are located in a file named *fft_scalar.f90*. Routines for 1D, 2D and 3D FFT are defined in this file. They serve as wrappers and invoke corresponding routines of the aforementioned numerical libraries, where the actual computation is performed. Selecting which particular numerical library will be used is performed by conditional compilation, using preprocessor directives (such as `#ifdef, #elif, #endif` and so on). Whenever a numerical library supporting FFT is found during the configuration phase of the QE software package, a corresponding macro parameter is defined in the Makefile, and is used to select an appropriate compilation path. For example, when the FFTW3 library is used, a macro parameter named `__FFTW3` will be defined, and only the code where FFTW3 routines are called will be compiled.

2.1 Enabling FFTE Library in Quantum ESPRESSO

We have extended QE to utilize the FFTE numerical library for performing FFT in 1D, 2D or 3D. The version of FFTE used is 5.0, accessible on the website [3]. FFTE is written in Fortran, supports parallelism with MPI, OpenMP, or both when hybrid variant is used. Also, FFT transformations for up to 3 dimensions are supported. Code development was done according to Quantum ESPRESSO development manual [4], which defines guidelines regarding the programming style.

A new macro parameter named `__FFTE` was created, and used in parts of the source code whenever a FFTE routine is called, or some initialization is performed. The configure script was also modified so that the configuration process can recognize if the FFTE library is present on the system, whether on the system path, or in the path specified during configuration. If the library is found, the `__FFTE` macro parameter is added to the Makefile. Variables needed to initialize FFTE, or store data between execution of FFTE routines were introduced as to be easily distinguishable by their prefix (`fftе_`).

In Quantum ESPRESSO, an internal decomposition of the data is used to perform 3D FFT transforms as a combination of multiple calls to serial 1D and 2D FFT routines, which are divided among processes. MPI is used for communication and data exchange in-between these phases. The reason for this approach is to avoid performing unnecessary transforms of subsections of the large 3D grid which already have zero values, as this pattern is common in data sets used by QE. A more detailed explanation of this decomposition can be found in Ref. [5].

It should be mentioned that the FFTE library does not support computation on many Fourier Transforms (on different arrays), in a single routine call. This can have some impact on the performance, because in QE there are many calls to 1D and 2D routines needed to complete transform on the entire data set. Also, when using FFTE, an initialization routine needs to be called before each transform, which includes even more overhead during execution. Significant drops in performance were not observed during our testing, but these factors should be considered when using the FFTE library in other projects.

2.2 Enabling FFTW3 Threading in Quantum ESPRESSO

Second extension of QE is related to support of threading of the FFTW3 library, which would enable hybrid parallelism (when used combined with the MPI), since it is already supported in Quantum ESPRESSO. The FFTW3 library supports threading in two modes:

- implicit, where an additional library `libfftw3_omp` has to be installed; in this case, FFTW3 routines support multi-threaded execution internally, so they are called like the serial ones, and
- explicit, where serial routines are used, but are called from within multiple threads running in parallel; this is possible because routines for FFT execution are thread-safe.

The following pseudocode representation roughly shows how the two threading modes were implemented in Quantum ESPRESSO (for the implicit mode, a single internally threaded routine call performs `nsl` transforms on arrays with length of `dim_z`, and for the explicit mode each routine call is serial):

- implicit

```
fftw_execute_many_dft(fw_plan, c, cout, nsl, dim_z)
```

- explicit

```
#pragma omp for
for i=1 to nsl
  offset=(i-1)*dim_z
  fftw_execute_dft(fw_plan, c[offset], cout[offset])
end for
```

The FFTW3 library supports reusing of plans, and also supports calculation of many transforms within a single routine call. This allows greater flexibility when using multiple transforms, and is optimal in terms of performance. More details on this can be found in Ref. [6].

In order to use implicit threading, FFTW3 thread initialization routines had to be called before calling any FFTW3 routines for FFT planning and execution. After the thread initialization has been successfully performed, the code for serial version can be reused, and threading is done automatically in the library routines.

With explicit threading, some modifications had to be made with the code. Because in the serial version many 1D or 2D transforms are aggregated in a single call for efficiency, execution had to be split into separate routine calls for each transform. This way, we actually had many routine calls, which can then be called from parallel threads. An OpenMP `parallel for` region was inserted, where in each iteration of the loop, FFT is performed on a separate sub-array. Since these routines are executed in parallel, and there are no data dependencies between loop iterations, this approach could be applied successfully.

3 Performance Tests

Here we will present performance tests done to compare newly supported FFTE library, and also performance of threaded FFTW3 library. Benchmarks were performed so that the performance was compared to most similar numerical libraries already supported in Quantum ESPRESSO.

3.1 FFTE Performance

We have tested Quantum ESPRESSO with enabled FFTE library, and compared it with the FFTW3 library that is already supported. These tests show only performances of serial libraries, since threaded FFTE was not implemented (because it wasn't always reliable when built with some compilers).

The cluster used for testing is made of nodes containing two AMD Magny-Cours Opteron 6174 processors, with 12 cores each. Nodes are connected via Infiniband network. The GCC compiler suite [7] was used in testing on this cluster. Our implementation was tested on benchmarks for PW module of Quantum ESPRESSO. FFTE Code was compiled with gfortran, version 4.1.2 with flags `-O3`, and the FFTW3 library was compiled with gcc, version 4.1.2 with flags

```
-O3 -fomit-frame-pointer -fstrict-aliasing -fno-schedule-insns
-ffast-math.
```

For the first test, up to 6 computing nodes were used (up to 72 processes). Execution times and scaling of the PW module are shown in Figure 1 for the case when the number of MPI processes is increased, and in Figure 2 when the problem size is increased, and the number of MPI processes stays constant (24 MPI processes were used in this test).

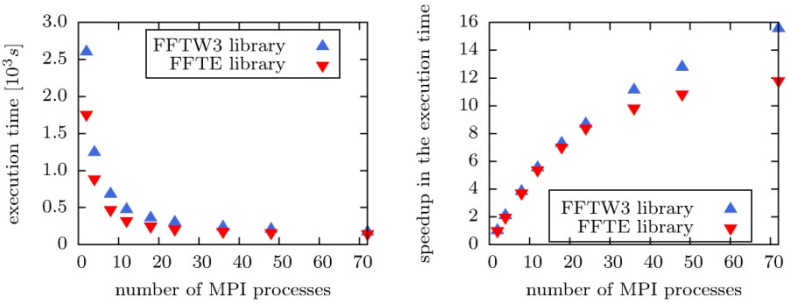


Fig. 1. Performance of the PW module of QE FFTE extension compared with the QE FFTW3 implementation: (left) Execution times of QE FFTW3/FFTE codes for different number of MPI processes; (right) Speedup in the execution time of QE FFTW3/FFTE codes as a function of a number of MPI processes (execution time on 1 MPI process used as a baseline)

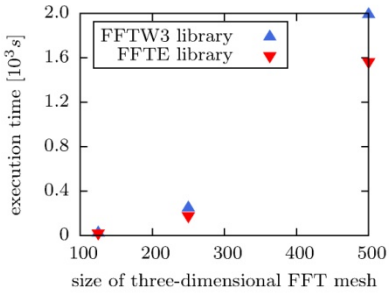


Fig. 2. Performance of the PW module of QE FFTE extension compared with the QE FFTW3 implementation: execution times as a function of 3D FFT mesh size

It can be seen that the FFTE library slightly outperforms FFTW3 in both cases (execution times are lower for the FFTE). The gap in performance grows as the size of the problem grows, so the FFTE seems suitable for large test cases. The difference in performance that is related to the problem size is also exhibited in the test case with the increasing number of MPI processes. As the number of MPI processes grows, each process gets less and less data to compute, and the difference in execution time diminishes. Because of this, the FFTE library shows worse speedup than the FFTW3.

3.2 FFTW3 Threaded Performance

For the performance testing of the threaded FFTW3 library, an FFTW (version 2) library internally supplied with Quantum ESPRESSO was selected for comparison. This was done because it was the only library supporting threading in

the hybrid mode (when used together with the MPI), and is also open-source and widely available.

Implementation of threaded FFTW3 was tested on a cluster with Intel Xeon processors, with two quad-core CPUs per node, and with Gigabit Ethernet interconnecting network. Library code was compiled with the Intel's icc compiler version 11.1 with the `-O3` flag, and Intel's ifortran was used for compilation of Quantum ESPRESSO.

Hybrid extension of the FFTW3 library was also tested with benchmarks for the PW module of QE. Tests were conducted again in the similar way, increasing the number of CPU cores in one case, and increasing input grid size in another. Configurations of 2 and 4 threads per MPI process were used, and also compared to the pure MPI case. Both threading variants (implicit and explicit) were tested with the FFTW3 library, and its performance is shown along with the internally supplied FFTW library (labeled as FFTW internal) in Figures 3 and 4. Total number of computing cores at some point is fixed, and is equal to a number of MPI processes times the number of threads per MPI process.

From this we see that both threading variants implemented for FFTW3 outperform the internal FFTW when executed with hybrid parallelism for most cases. Although, both threaded libraries are still slower than the pure MPI version. This is probably due to the fact that for the type of input data used with Quantum ESPRESSO, the overhead related to the thread management is probably greater than benefits of reduced MPI communication. Evidence for this are runs with four threads per MPI process, where performance gets significantly worse.

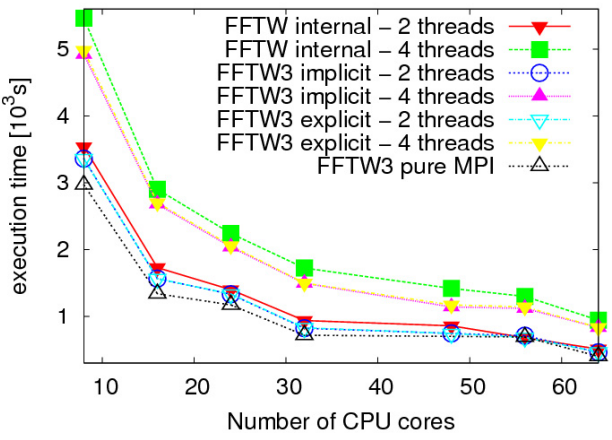


Fig. 3. Performance of the PW module of QE FFTW3 threaded extensions compared with the internal QE FFTW hybrid implementation and pure MPI FFTW3 implementation: Execution times of QE FFTW3 implicit and explicit/internal FFTW/pure MPI codes for different number of MPI processes

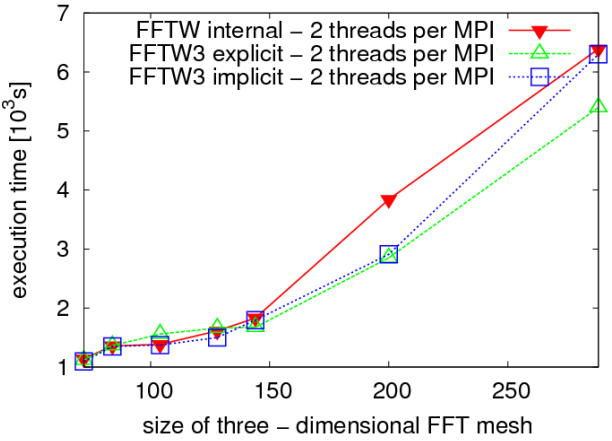


Fig. 4. Performance of the PW module of QE FFTW3 threaded extensions compared with the internal QE FFTW hybrid implementation and pure MPI FFTW3 implementation: QE FFTW3 implicit and explicit / internal FFTW execution times as functions of 3D FFT mesh size.

These results agree with what was presented in Ref. [5], where similar thing was investigated, and was shown that threading does not increase performance in all cases. Better performance was observed only in some cases where the number of MPI processes was significantly large. Also, Quantum ESPRESSO has other ways to control parallelism in software (for example, task groups, pools of processes, etc.) which is related to a particular input data set. Because these options were not primarily designed with hybrid parallelism in mind, it is not easy to fine tune Quantum ESPRESSO to achieve optimal performance when threading is used.

It is also worth mentioning that no significant difference in performance between implicit and explicit variants of FFTW3 threading was noticed. Looking at how threading is implemented in those two cases, an advantage of the explicit mode is that the OpenMP parallel region is created only once, and inside of it there are calls to many routines where FFT is computed. This should be optimal with regards to the overhead related to thread creation and synchronization. On the other hand, when using implicit threading, a new OpenMP parallel region has to be created with every routine call. However, because an advanced FFTW3 interface is used with implicit threading mode, it allows many transforms on different arrays to be aggregated in a single routine call from FORTRAN. It is possible that the native implementation of FFTW3 threaded library is aware of that, and that it successfully avoids unnecessary creation of parallel regions for each separate Fourier Transform.

4 Conclusions

In this project two extensions to Quantum ESPRESSO were implemented: the support for FFTE library for computing Fourier Transform in the serial mode, as well as the FFTW3 library in threaded mode. These extensions showed better performance compared to default QE libraries (open-source FFTW version 2 and 3 were selected for comparison). In the case of the FFTE library, performance increase could be significant when the large charge density mesh is requested for the simulation of a physical system. Both the explicit and implicit variants of FFTW3 threading showed better performance compared to internally supplied FFTW (version 2) when tested in hybrid configuration (two and four threads per MPI process), and while still not faster than the pure MPI version, should be considered when there is a need for hybrid parallelism. It is expected that a much larger problem size and more CPU cores are needed in order to get satisfactory performance of the hybrid FFTW3, which can match, or even surpass the performance of the pure MPI version.

Acknowledgements. Numerical results were obtained on the PARADOX cluster at the Scientific Computing Laboratory of the Institute of Physics Belgrade and on the NIIFI SC in Hungary. We acknowledge the support by the Serbian Ministry of Education, Science and Technological Development under projects No. ON171017 and III43007, and by the European Commission under FP7 projects HP-SEE, PRACE-2IP, PRACE-3IP and EGI-InSPIRE.

References

1. Quantum ESPRESSO official web site, `http://www.quantum-espresso.org`
2. Frigo, M., Johnson, S.G.: The Design and Implementation of FFTW3. Proceedings of the IEEE 93, 216 (2005)
3. FFTE: A Fast Fourier Transform package, `http://www.ffte.jp/`
4. Developer's Manual for Quantum ESPRESSO, `http://www.quantum-espresso.org/?page_id=47`
5. Spiga, F.: Implementing and Testing Mixed Parallel Programming Model into Quantum ESPRESSO. In: Science and Supercomputing in Europe - research highlights 2009, CINECA Consorzio Interuniversitario, Bologna (2010)
6. FFTW3 advanced interface, `http://www.fftw.org/doc/Advanced-Interface.html`
7. GCC compiler suite, `http://gcc.gnu.org/`

An Analysis of FFTW and FFTE Performance

Miloš Nikolić, Aleksandar Jović, Josip Jakić, Vladimir Slavnić, and Antun Balaž

Scientific Computing Laboratory, Institute of Physics Belgrade,
University of Belgrade, Pregrevica 118, 11080 Belgrade, Serbia
{milos.nikolic,aleksandar.jovic,josip.jakic,
vladimir.slavnic,antun.balaz}@ipb.ac.rs

Abstract. One of the most frequently used algorithms in engineering and scientific applications is Fast Fourier Transform (FFT). Its open source implementation (Fastest Fourier Transform of the West, FFTW) is widely used, mainly due to its excellent performance, comparable to the vendor-supplied libraries. On the other hand, even if not yet in a fully production state, FFTE (Fastest Fourier Transform of the East) keeps up with FFTW, and outperforms it for very large transform sizes. Here we present results of the performance and scalability tests of FFTW and FFTE libraries. Comparison is done using different compilers and parallelization approaches on CURIE and JUGENE supercomputers.

Keywords: FFT, MPI, OpenMP, Hybrid parallelism.

1 Introduction

The Discrete Fourier Transform (DFT) plays an important role in many scientific and technical applications, including time series and waveform analysis, solutions to linear partial differential equations, convolution, digital signal processing, and image filtering. The DFT is a linear transformation that maps n regularly sampled points from a cycle of a periodic signal, like a sine wave, onto an equal number of points representing the frequency spectrum of the signal. In 1965, Cooley and Tukey [1] devised an algorithm to compute the DFT of an n-point series in $n*\log(n)$ operations. Their new algorithm was a significant improvement over previously known methods for computing the DFT, which required n^2 operations. The revolutionary algorithm by Cooley and Tukey and its variations are referred to as the Fast Fourier Transform (FFT). Due to its wide application in scientific and engineering fields, there has been a lot of interest in implementing FFT on parallel computers.

The scalability of 3-dimensional Fast Fourier Transforms (3D FFTs) is limited by the all-to-all nature of the communications involved. It presents a challenge scaling up those codes that rely heavily on FFT methods to exploit existing and future Petascale supercomputing systems.

The goal of this paper is to assess the performance and scalability of various implementations of FFT. Specific FFT benchmark codes were developed and used to compare performance of different 3D FFT library routines and explore their scalability in the strong sense on CURIE [2] and JUGENE [3] supercomputers, provided by PRACE [3] association.

M. Dulea et al. (eds.), *High-Perf. Comp. Infrastr. for South East Europe's Research Communities*, Modeling and Optimization in Science and Technologies 2,
DOI: 10.1007/978-3-319-01520-0_20, © Springer International Publishing Switzerland 2014

In section 2, we introduce various FFT methods used in this study. In section 3 we give a description of benchmarking procedures for FFT libraries using in-house developed FFT test codes. Section 4 presents benchmarks results and, finally, in section 5, we summarize our conclusions, discuss related work, and make some recommendations.

2 FFT Libraries and Methods

The main performance bottleneck of parallel 3D FFTs is the communication. Once 3D data is distributed over MPI processes, all-to-all communications are unavoidable. Applications that rely on FFTs adopt different data decomposition strategies: 1D decompositions give each process a complete 2D slab, 2D decompositions give each process a complete 1D pencil, while 3D decompositions give each process a block that does not span the global domain in any dimension. Slab decompositions tend to perform well on small process counts; pencil decompositions scale better, but also eventually run out of steam. Efforts to optimize the performance of 3D parallel FFT libraries have tended to focus on slab and pencil decompositions.

2.1 FFTW

The "Fastest Fourier Transform in the West" has been developed at Massachusetts Institute of Technology by Matteo Frigo and Steven G. Johnson [5]. It is open source and free library written in C, but also has Fortran bindings. It supports transforms of arbitrary sizes. The performance of FFTW is competitive with, and sometimes exceeds, vendor-supplied libraries, and has the advantage that the library and its performance are both highly portable. FFTW achieves portable performance by measuring the speed of many alternative codelets on the target architecture, and making an informed choice at run-time.

Results in this study were obtained using release 3.3.1 of FFTW, the first version to support parallel MPI 3D FFTs. Only slab decompositions are currently supported, so that the 3D grids are decomposed in only one dimension (here we use the z coordinate).

2.2 FFTE

FFTE [6] has been developed by Daisuke Takahashi of Tsukuba, Japan. The name FFTE, which is an acronym for "Fastest Fourier Transform in the East", is more of a tribute to FFTW than a signal of any serious attempt to offer a production-ready library to rival FFTW (even though FFTE has been observed to slightly outperform FFTW on very large FFTs). FFTE supports radix 2, 3, and 5 Discrete Fourier Transforms (DFTs), including optimised routines for radix 8, and has parallel flavours, both pure MPI and hybrid (MPI/OpenMP). FFTE comes with little documentation, and it is necessary to examine the source code in order to use it. The MPI-parallel version only works correctly when the number of MPI processes is a power of 2. In other cases but

the results will be invalid but the program would run nevertheless, so you should be carefull. In FFTE, 3D parallel FFTs must be decomposed over MPI processes so that the leading coordinate (x) of the 3D arrays (x, y, z) is kept local to each MPI process.

In this study, we used version 5.0 of FFTE. We employed both PZFFT3D, a parallel 3D DFT method which requires that the data is decomposed over MPI processes in the z-coordinate (i.e. it supports only a slab decomposition), and PZFFT3DV, which allows data decomposed in both the y and z coordinate (i.e. it supports a pencil decomposition). Both PZFFT3D and PZFFT3DV will utilize any additional OpenMP threads, if available at run-time.

FFTE uses MPI_ALLTOALL to implement the MPI communication phases in both PZFFT3D and PZFFT3DV.

3 Benchmarking of FFT Libraries Using Developed In-House Codes

For the purpose of performance and scalability testing of various FFT libraries, in-house benchmark codes were developed on a local PARADOX cluster at the Institute of Physics Belgrade (IPB) using C, Fortran77 and Fortran90 programming languages and the latest versions of FFTW (3.3.1) and FFTE (5.0) libraries, at the time. Since the FFTE package is distributed with Fortran source files only, a suitable FFT library was created. A comparison of FFTW and FFTE libraries was performed on CURIE and JUGENE for different types and dimensions of FFT calls (MPI and hybrid with MPI/OpenMP) and we have chosen to use 3D hybrid benchmark codes among them as the most relevant. Obtained measurement results on these codes are presented.

CURIE is located in the computing center of CEA (TGCC) at Bruyères-le-Chatel in France. We used BULLX Fat nodes which have four eight-core Intel Nehalem-EX X7560 processors with 128 GB of memory. JUGENE is located in The Jülich Supercomputing Centre in Germany. It is based on IBM BlueGene/P architecture with four PowerPC 450 32-bit cores and 2 GB of memory in each compute node.

On CURIE, hybrid tests were performed with the number of threads per MPI process varying from 4 to 32 and for the total number of cores ranging from 32 to 1024. On JUGENE, hybrid tests were performed using 1-4 threads per MPI process using 16 to 512 total cores. FFT testing was performed on complex array of varying sizes (up to 230). Input datasets were chosen to be comparable with the ones used in FFTW and FFTE developers test examples, both in size and operational complexity. In order to allow detailed performance analysis of the execution time of our implementation, the forward FFT was looped (in-place) 120 times on CURIE and 1000 times on JUGENE.

4 FFT Benchmark Codes Results and Interpretation

Using the in-house developed FFT benchmark code, we have compared the execution times of the considered libraries for 3D Fourier transform computation of the 3D mesh with dimensions 1024^3 on CURIE and 256^3 on JUGENE (due to the memory limitations of the JUGENE nodes, a smaller grid was used in this case).

4.1 CURIE Results

As presented in Fig. 1 the FFTW 3.3.1 library demonstrates better scalability than FFTE, but FFTE performs faster (achieves lower execution times) than FFTW when pure MPI implementations are compared on CURIE.

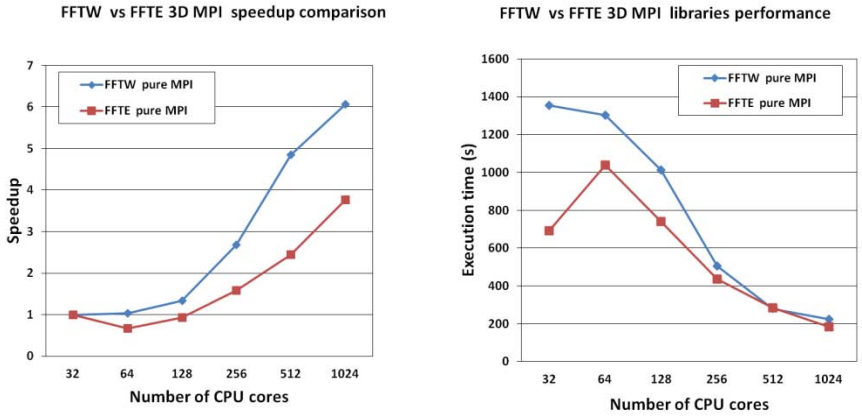


Fig. 1. Comparison of FFTE and FFTW pure MPI performance for 1024^3 dataset on CURIE: (left) speedup plot (32 cores execution times used as a baseline); (right) execution times plot

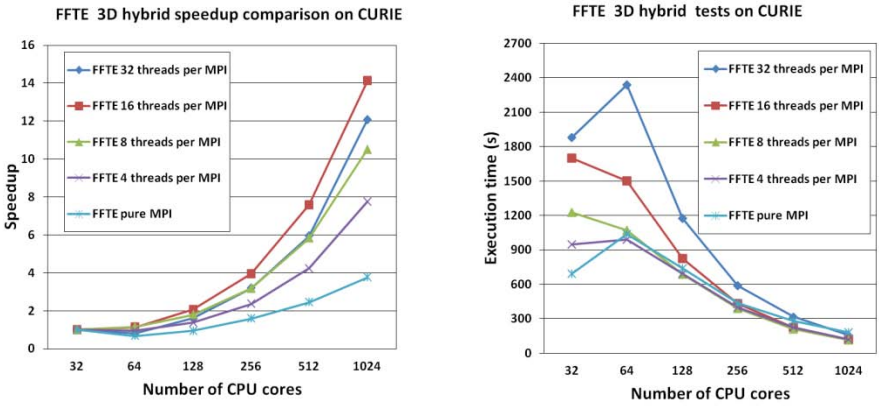


Fig. 2. Comparison of FFTE hybrid performance for different MPI/threads combinations using 1024^3 dataset on CURIE: (left) speedup plot (32 cores execution times used as a baseline); (right) execution times plot

Figure 2 shows that the best scaling is achieved when running with 16 threads per MPI process and that the fastest hybrid combination is the one with 4 threads per MPI process. From this figure we can also see that the FFTE library implemented with pure MPI scales worse than the hybrid implementation for all tested combinations of

processes and threads. However, Fig. 2. (right) shows absolute execution times, and we see that tests performed with pure MPI are faster than hybrid tests with both 32 and 16 threads per MPI process, and are comparable to hybrid runs with 4 and 8 threads per MPI process. As it can be observed, execution times for threaded runs increase as the number of threads per MPI process increases. This can be due to overheads related to the thread initialization and management, but also due to different ways memory allocation is performed in NUMA environment with a process-oriented configuration (MPI) and a thread-oriented configuration (OpenMP).

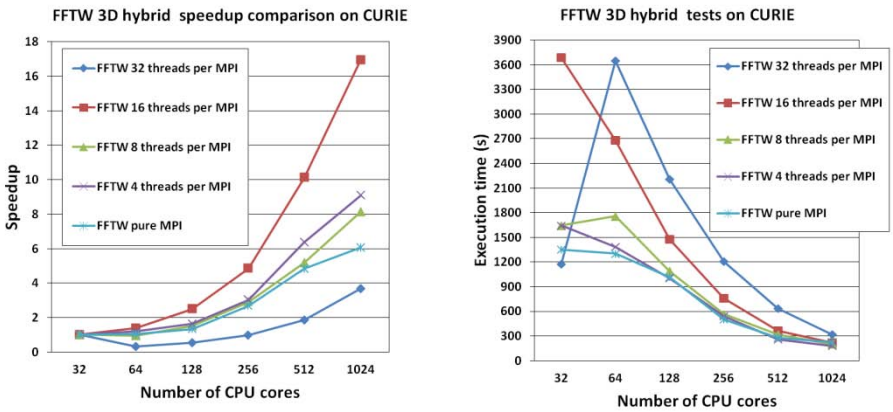


Fig. 3. Comparison of FFTW hybrid performance for different MPI/threads combinations using 1024^3 dataset on CURIE: (left) speedup plot (32 cores execution times used as a baseline); (right) execution times plot

Figure 3 shows hybrid tests for the FFTW library with 4, 8, 16 and 32 threads per MPI processes. The tests performed on CURIE show that the best scaling is achieved when running with 16 threads, as in the case of the FFTE library. Also, the fastest hybrid combination is the one with 4 threads, the same as in the case of FFTE library. Pure MPI results are shown for comparison and it can be seen that pure MPI results are comparable with the fastest hybrid implementation. We have observed unusual performance for the case of a single MPI process and 32 threads, where performance is significantly better. This is probably due to the internal implementation of the hybrid version of the library, and this case needs further investigation using appropriate tools.

Figure 4 shows that FFTE library performs faster than FFTW for all hybrid combinations, which were tested on 512 and 1024 cores on the CURIE machine.

Apart from the case with the total of 32 cores, both the MPI and hybrid versions show very similar performance, with hybrid versions performing slightly faster as the number of cores grows (clearly visible in the case of the FFTE library). Because of that, we recommend using a hybrid implementation when the total number of cores is sufficiently large.

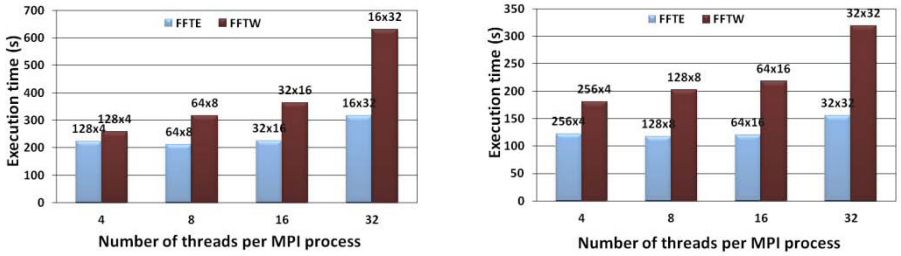


Fig. 4. Comparison of FFTE and FFTW hybrid performance on CURIE: (left) 512 total cores; (right) 1024 total cores

4.2 JUGENE Results

Figure 5 shows that again FFTW 3.3.1 library scales better than FFTE, but the FFTE library is faster than FFTW in absolute execution times when implemented with pure MPI on JUGENE.

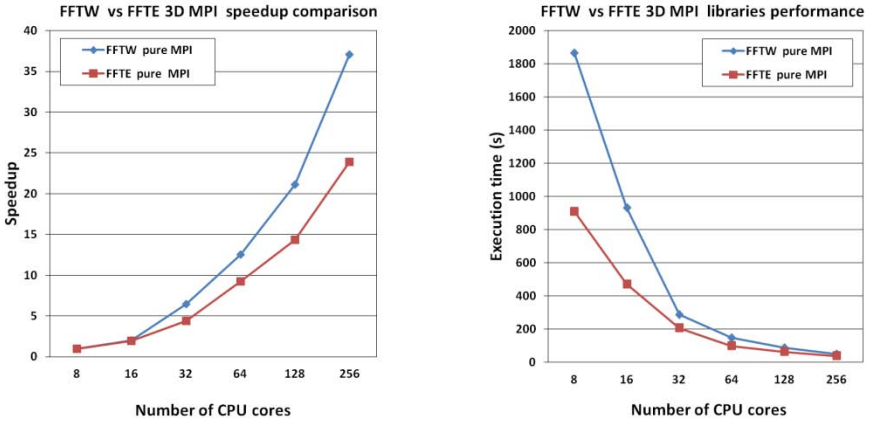


Fig. 5. Comparison of FFTE and FFTW pure MPI performance for the 256^3 dataset on JUGENE: (left) speedup plot (8 cores execution times used as a baseline); (right) execution times plot

Figure 6 presents hybrid tests for the FFTE library with pure MPI, as well as for two and four threads per MPI process. The tests performed on JUGENE show that better scaling is achieved when four threads are used. However, again in Fig. 6 (right) we see that the pure MPI implementation is the fastest.

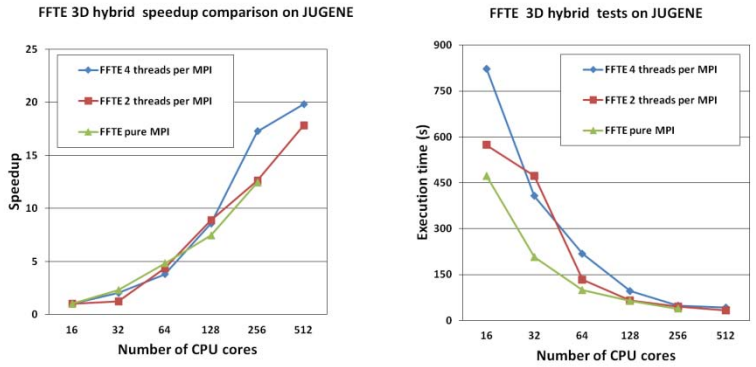


Fig. 6. Comparison of FFTE hybrid performance for different MPI/threads combinations using 256^3 dataset on JUGENE: (left) speedup plot (16 cores execution times used as a baseline); (right) execution times plot

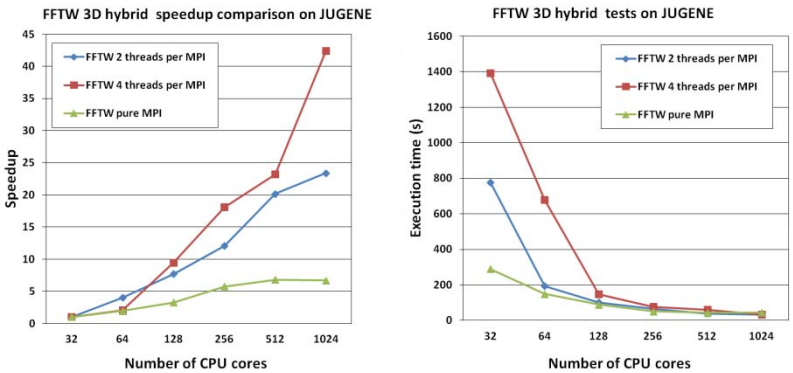


Fig. 7. Comparison of FFTW hybrid performance for different MPI/threads combinations using 256^3 dataset on JUGENE: (left) speedup plot (32 cores execution times used as a baseline); (right) execution times plot

Fig. 7 shows the hybrid tests for FFTW 3.3.1 library with pure MPI, as well as for two and four threads per MPI process. Tests performed on JUGENE show that better scaling is achieved when running with four threads than with two threads per MPI process. It is interesting to notice that in both cases this library shows excellent scaling on the JUGENE system. Fig. 7 (right) shows that the FFTW library is faster for tests with two threads per MPI process than tests with four threads in all cases, but that the pure MPI implementation outperforms all others.

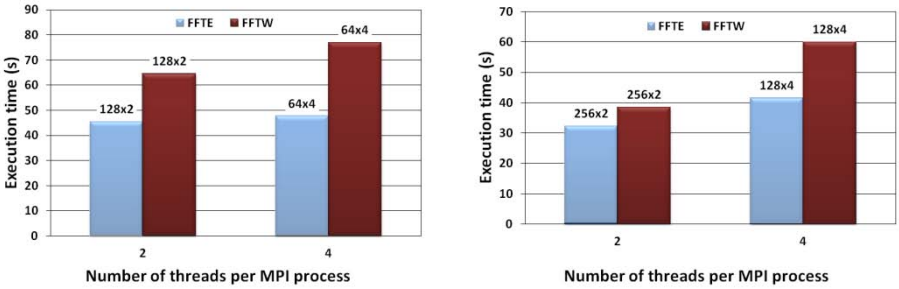


Fig. 8. Comparison of FFTE and FFTW hybrid performance on JUGENE: (left) 256 cores; (right) 512 cores

5 Conclusions and Recommendations

The scalability of parallel 3D FFTs remains inherently limited, owing to the all-to-all communications involved. Likewise, the variety of data decompositions supported by the available libraries is also limited. . Given the current state of affairs, it is difficult for application developers to rely on third party libraries to achieve portable and scalable FFT performance. While these limitations of numerical library routines remain to be the case, we will continue to see FFT-dependent applications using custom parallel FFTs with bespoke communications, and little re-use of library code, often restricted to serial or threaded FFTs within a single MPI process.

It is clear that exploiting benefits of shared memory within a node can help improve the scalability and for this reason using a hybrid implementation, when the total number of cores is sufficiently large, is recommended.

Acknowledgements. The work is achieved using the PRACE Research Infrastructure resources [BULL Bullx (CURIE), France; Blue Gene/P (JUGENE), Germany] and PARADOX Cluster at the Scientific Computing Laboratory of the Institute of Physics Belgrade, supported in part by the Serbian Ministry of Education, Science and Technological Development under projects No. ON171017 and III43007, and by the European Commission under FP7 projects HP-SEE, PRACE-2IP, PRACE-3IP and EGI-InSPIRE.

References

1. Cooley–Tukey FFT algorithm, `http://en.wikipedia.org/wiki/Cooley-Tukey_FFT_algorithm`
2. CURIE Supercomputer, `http://www-hpc.cea.fr/en/complexe/tgcc-curie.htm`
3. JUGENE Supercomputer, `http://www.fz-juelich.de/jsc/jugene`
4. PRACE Home Page, `http://www.prace-ri.eu/`
5. FFTW Home Page, `http://www.fftw.org/`
6. FFTE: A Fast Fourier Transform Package, `http://www.ffte.jp/`

Author Index