

IPPT Reports on Fundamental Technological Research

3/2014

Eligiusz Postek, Tomasz A. Kowalewski, *Editors*

**GRAFEN – IPPT PAN COMPUTER
OF BIOCENTRUM OCHOTA GRID**

Institute of Fundamental Technological Research
Polish Academy of Sciences

Warsaw 2014

IPPT Reports on Fundamental Technological Research

ISSN 2299-3657

ISBN 978-83-89687-91-3

Editorial Board/Kolegium Redakcyjne:

Wojciech Nasalski (Editor-in-Chief/Redaktor Naczelny),
Paweł Dłużewski, Zbigniew Kotulski, Wiera Oliferuk,
Jerzy Rojek, Zygmunt Szymański, Yuriy Tasinkevych

Reviewer/Recenzent:

Tomasz A. Kowalewski

Received on 1st December 2014

Copyright © 2014 by IPPT-PAN

Instytut Podstawowych Problemów Techniki Polskiej Akademii Nauk (IPPT-PAN)
Institute of Fundamental Technological Research Polish Academy of Sciences
Pawińskiego 5B, PL 02-106 Warsaw, Poland

Printed by/Druk:

EXPOL, P. Rybiński, J. Dąbek, Sp. J., ul. Brzeska 4, 87-800 Włocławek, Poland

This work was prepared in the frame of the project:

“Biocentrum Ochota – computational infrastructure for strategic development of the biology and medicine”,

co-funded by the European Regional Development Fund as part of the Innovative Economy program.



Acknowledgements

This publication was made possible by the input of the following persons:

Jerzy Supel, Mikołaj Dybko, Piotr Putek, Przemysław Pindakiewicz, Maciej Kowalczyk, Arkadiusz Grubba, Piotr Przybyła, Dorota Sobczyńska and several others.

GRAFEN – IPPT PAN COMPUTER OF BIOCENTRUM OCHOTA GRID

Eligiusz Postek, Tomasz A. Kowalewski, *Editors*

Institute of Fundamental Technological Research
Polish Academy of Sciences

Abstract

The report treats on computing cluster in the IPPT. The first part of the report is a user manual of the cluster. In the following, we described chosen applications concerning applications of numerical methods in the computational biology, tectonophysics and fluid mechanics. Finally, we depicted a few selected computer programs that are implemented on the cluster.

GRAFEN – IPPT PAN COMPUTER OF BIOCENTRUM OCHOTA GRID

Pod redakcją:
Eligiusza Postka i Tomasza A. Kowalewskiego

Instytut Podstawowych Problemów Techniki
Polskiej Akademii Nauk

Streszczenie

Praca traktuje o sposobie użytkowania klastra obliczeniowego znajdującego się w IPPT. Praca zawiera podręcznik użytkownika. W dalszym ciągu zostały opisane wybrane aplikacje dotyczące zastosowań metod numerycznych w problemach biologii obliczeniowej, tektonofizyki i mechaniki płynów. Na zakończenie podane zostały przykłady wybranych implementowanych programów komputerowych wraz z ich krótkimi opisami.

Contents

1. Preface	11
2. Supercomputers, <i>M. Milewska</i>	15
2.1. History	15
2.2. TOP 500 List.....	18
2.2.1. Top 10 Supercomputers	19
2.3. Supercomputers in Poland	27
2.3.1. Polish TOP500.....	29
2.3.2. New Polish investments.....	33
2.4. Future of supercomputers.....	34
2.5. Short history of IPPT computational resources.....	35
3. Grid computing, <i>R. Gawlik</i>	37
3.1. Differences between grid and cluster	37
3.2. Examples of grid systems in Poland	38
3.3. Grid BIOCENTRUM OCHOTA	39
3.4. Grafen cluster.....	39
3.4.1. Equipment.....	39
3.4.2. Software	42
4. Scientific software, <i>R. Gawlik</i>	45
4.1. ABAQUS	45
4.1.1. ABAQUS/Standard.....	45
4.1.2. ABAQUS/Explicit	45
4.1.3. ABAQUS/CAE.....	46
4.1.4. Links	46
4.2. ANSYS CFD.....	46
4.2.1. ANSYS Fluent	46
4.2.2. ANSYS CFX.....	47
4.2.3. Workbench.....	47
4.2.4. Links	47
4.3. COMSOL	48
4.3.1. Links	48

4.4. DL_POLY CLASSIC	49
4.4.1. Links	49
4.5. LAMMPS	50
4.5.1. Links	50
4.6. LMGC90	51
4.6.1. Links	51
4.7. LS-DYNA	51
4.7.1. Links	52
4.8. MATHEMATICA.....	52
4.8.1. Links	53
4.9. MATLAB	53
4.9.1. Toolbox.....	54
4.9.2. Parallel programming	54
4.9.3. Links	55
4.10.MUMPS	55
4.10.1. Links	55
4.11.OPENFOAM.....	56
4.11.1. Links	56
4.12.PARAVIEW.....	57
4.12.1. Links	57
4.13.TECPLOT 360	57
4.13.1. Links	57
4.14.VMD	58
4.14.1. Links	58
5. Grafen cluster user's guide, R. Gawlik, E. Postek	61
5.1. Introduction.....	61
5.2. How to use the system	62
5.3. Virtual servers.....	62
5.3.1. Windows 2008 server	62
5.3.2. MATLAB Computing Server	63
5.4. HPC system access	63
5.4.1. LOG IN – SSH.....	63
5.5. Files transfer – scp	64
5.5.1. Linux and Mac OS.....	64
5.5.2. Windows	64
5.6. Files transfer – sftp.....	64
5.6.1. Interactive mode	64
5.6.2. Batch mode	65
5.7. Task manager	65
5.7.1. Available resources	65
5.7.2. Starting the task	66
5.7.3. Accurate resources identification.....	68
5.7.4. Parallel programs	69
5.8. Task monitoring	71

5.8.1. Task state	71
5.8.2. Deleting the task	74
5.8.3. Output data.....	74
5.9. Environment modules	74
5.10. Mass storage.....	77
6. Starting applications on cluster, R. Gawlik	79
6.1. Introduction.....	79
6.2. ANSYS Fluent Example	79
6.3. COMSOL Example.....	81
6.4. LS-DYNA Example.....	82
6.5. MATLAB Example	84
6.6. Environment module programs.....	85
7. Selected applications	87
7.1. Parameter sensitivity of a tensegrity model of a tissue, <i>E. Postek</i>	87
7.1.1. Introduction.....	87
7.1.2. Methodology, mechanical model.....	88
7.1.3. Mathematical formulation.....	89
7.1.4. Computer implementation	92
7.1.5. Numerical illustration	94
7.1.6. Final remarks	97
7.2. Generic models of linear and non-linear visco-elastic surface deformation, <i>E. Postek</i>	98
7.2.1. Problem statement.....	98
7.2.2. Nonlinear creep.....	99
7.2.3. Numerical algorithm	101
7.2.4. Numerical example	102
7.2.5. Final remarks	105
7.3. Microstructure-based modelling of sound absorption in rigid porous media, <i>T. Zieliński</i>	106
7.4. Reliability analysis of reinforced concrete structures, <i>E. Postek</i>	111
7.4.1. Introduction.....	111
7.4.2. Reliability algorithm	111
7.4.3. Performance functions	112
7.4.4. Parameter sensitivity.....	113
7.4.5. Constitutive model.....	114
7.4.6. Applied finite element.....	116
7.4.7. Computer program.....	122
7.4.8. McNeice plate.....	123
7.4.9. Nuclear containment shell.....	124
7.4.10. Final remarks	131
7.5. Development of a concept of an agent-stress model of a tissue, <i>E. Postek</i>	133

7.5.1. Problem statement.....	133
7.5.2. General algorithm.....	134
7.5.3. Characterization of the agent model	134
7.5.4. Agent model.....	135
7.5.5. Coupling scheme.....	136
7.5.6. Mechanical model.....	137
7.5.7. Summary.....	139
7.6. Mould filling with solidification, <i>E. Postek</i>	140
7.6.1. Introduction.....	140
7.6.2. Flow problem.....	140
7.6.3. Microstructural solidification model.....	141
7.6.4. Numerical example	143

Appendix **147**

A.1. FibFlex – Computation of flexible polymer nanofibers bending.....	147
A.2. SpatKin – rule-based modeling of spatially extended biochemical systems	148
A.3. DrugRelease	149
A.4. AceGen/AceFEM	151
A.5. Crack modeling in composites package	152
A.6. Residual Thermal Stress (RTS) Package.....	153
A.7. YADE-PM – Extension of YADE discrete element method environment.....	154
A.8. Nanotrack: Nanoparticles motion detection program.....	155
A.9. Original computer program MC-UNBAL	156
A.10. STAND – STochastic ANalysis and Design	157
A.11. DCAST	158
A.12. DMFILL	159
A.13. DEMSYS	160
A.14. DTENS	161
A.15. OREGANO-VE.....	162
A.16. VECDS – Visual Editor of Crystal Defects.....	163

Preface

The IPPT PAN takes part in the project “Biocentrum Ochota – computational infrastructure for strategic development of the biology and medicine”. The main goal of the project is the creation of the computational infrastructure of the consortium Biocentrum Ochota. The infrastructure should allow creation and integration of the databases and applications basing on existing, unique, computing services.

There are six institutions taking part in the project:

- Institute of Biochemistry and Biophysics Polish Academy of Sciences (the project coordinator)
- Nałęcz Institute of Biocybernetics and Biomedical Engineering Polish Academy of Sciences
- Nencki Institute of Experimental Biology Polish Academy of Sciences
- Mossakowski Medical Research Centre Polish Academy of Sciences
- Institute of Fundamental Technological Research Polish Academy of Sciences
- The International Institute of Molecular and Cell Biology

The consortium Biocentrum Ochota possesses a unique in Poland scientific potential in the field of state-of-art biological research and applications of the research into medicine and biotechnology. As a result of their research, the institutes belonging to the consortium have created numerous bioinformatics methods that are interesting for other research institutions, education, health service and high-tech companies. The cooperation of the institutes within Biocentrum Ochota creates an extraordinary possibility of setting up an integrated infrastructure with respect to the functionality and the hardware that allows for access to the applications and the databases. The access to the infrastructure has been limited so far, both due to shortage of the equipment and the possibility of cooperation in a proper range.

The completing of the project has provided the development of the research methodologies and better experimental data interpretation in the widely understood biomedicine and biotechnology. The synergy of the competencies of the participants of the project allowed for proposals of a range of services. They are dedicated to numerous groups of the potential users, namely, physicians, entrepreneurs, academic teachers, students and research scientists. In contrast to the benefits, the relatively small financial effort put on the project arises from the employment of the already existing potential accumulated in the institutions belonging to the consortium.

The integration of the already existent computational tools is designed in such a way that all instruments that are provided by the particular institutions are available via thematic portals (BioInfo, BioMed, BioTech). The portals are addressed to the user's groups (science, education, health service, industry) without the restraints of computational resources. The necessary hardware and software were provided, installed and put into operation.

The system integrated distinctive services of the data processing and analytical devices belonging to the members of the consortium. The hardware base for the applications consists of six clusters of total computational efficiency 4 TFLOPS. The clusters are bonded with fast connections. The total disk space is about 4 Petabytes. The network allows for access to virtual workstations of on-the-fly user-defined resources. This all stands for GRID type network. The entire system is accessible via dedicated graphic workstations from the PC's or workstations that are connected locally in the institutional networks. The services can be used for the users who are outside of the institutions. The access for them is provided via VPN connections that are set in the local networks of the institutions. A part of the services is available from tablets and mobile phones.

In the case of IPPT PAN the HPC cluster named "Grafen" has been implemented along with several applications and services. The cluster is the part of the Biocentrum Ochota grid fulfilling the assumptions of the project.

The main role of the "Grafen" cluster for the researchers in the IPPT is a bridge for High Performance Computing. All of the HPC machines have a few common features. The features are a large number of processors, the large amount of memory, Unix/Linux type operation systems and queuing systems. The usage of unix/linux operation systems family allows for quite smooth passing from local PC-computers, workstations and laptops to work on HPC machines. However, the condition to work on them is the parallel structure of the program we wish to transfer to HPC machine. In the most cases, the programmer develops his application with a low number of cores. A few cores are available on good laptops, recently. However, again, to develop and test the application for a higher number of cores the programmer needs a cluster of processors. The cluster in IPPT serves as a platform for developing applications,

and executing smaller production runs. The cluster is used for training purposes during a course on postgraduate studies in IPPT PAN as well.

The report consists of three main parts. The first part describes the main features of the cluster, the implemented tools for the software developers, the user's handbook and a variety of software applications belonging to both commercial and public domain groups. The second part treats on applications that have been already done, under development or presenting the offer to the potential users. The third part of the report deals with the description of the software that is being developed or has already been developed by the researchers from IPPT PAN. This all creates the system of services for research teams in the Biocentrum Ochota and beyond.

Tomasz A. Kowalewski
Eligiusz Postek

Supercomputers

Monika Milewska

Usually supercomputers are understood as computers well above the capabilities of computers commonly used. This term appeared in the '60s with respect to computers manufactured by CDC and later by Cray. They were produced in dozens of copies and costed several million dollars. Today, most supercomputers are made in single copies, designed and custom-made, usually using mass-produced processors and other components. The costs of their production back to a billion dollars.

The word “supercomputer” is very often associated with the IBM supercomputer named Deep Blue which defeated grandmaster Gary Kasparov in 1997 and became the most well-known public face of supercomputers.

But in fact, supercomputer is any computer that's one of the most powerful, fastest systems in the world at any **given point in time**. Multiple processors were linked together and multiple processes could be performed on the same sets of data. More than one central processing unit (CPU) and enormous amount of storage allows to accomplish more tasks at once and can access many tasks simultaneously because of its capability to do vector arithmetic.

Supercomputers are necessary in nowadays world to solve problems of science and industry which are closely related to economic growth. Their ability to perform computations very rapidly, handle extremely large volumes of data and simulate and model results makes them priceless for a wide range of computationally intensive tasks of research projects in various field such as: quantum mechanics, weather forecasting, climate research, oil and gas exploration, molecular modelling, physical simulations.

2.1. History

The history of computer-aided solution problems goes back to the early 1920s but the evolution of computers accelerated in the 40s, when the first computing machines were used for military purposes during the second World War. For example British computer named Colossus, designed to read German Enigma encoded information,

could read up to 5,000 characters per second. First programmable and fully operational electromechanical German computer Z3 were used, e.g. to design the geometry of a German aircraft wing. ENIAC (Electronic Numerical Integrator And Computer) the first electronic general-purpose computer was created in USA, also at this time. It was Turing-complete, digital, and capable of being reprogrammed to solve a large class of numerical problems. ENIAC contained about 18 thousand vacuum tubes and 1,500 relays. It weighted 27 t, took up 1800 square feet (167 m²), and consumed 150 kW of electric power. ENIAC had no RAM memory and yet it has been used mainly for calculations relating to ballistics, production of nuclear weapons, weather forecasting, design and wind tunnel study of cosmic rays. It was also used to study and analyze random rounding errors.

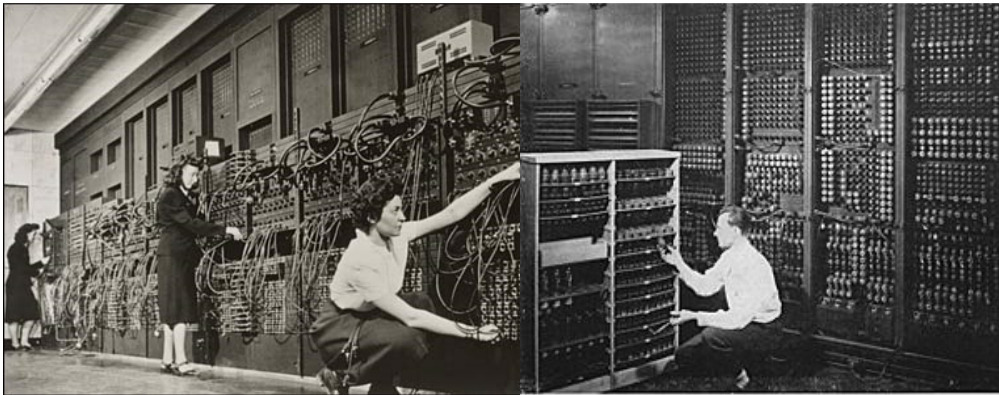


Figure 2.1.1. ENIAC supercomputer at University of Pennsylvania's Moore School of Electrical Engineering.

The new generation of supercomputers based on semiconductors appeared in to the late 1960s and 1970s. The supercomputers were used to utilize vector processing techniques which altered the way the data was processed and speeded it up. They also used the memory in form of magnetic cards, had integrated circuits and the transistor developed in MOS and LSI technology. The first supercomputer Atlas was designed in England in 1962. It was considered to be the most powerful computer in the world at that time, operating at a rate of 1 million operations per second. It's memory structure is considered as the emergence of a new type of memory – associative memory, which evaluated later to contemporary RAM memory. In Poland, the construction of Odra 1003 began in early 1960s, and its serial production began in 1964. In that time, IBM created the first personal computers with its own operating system.

Supercomputers of the 1980s used only a few processors, in the 1990s, machines with thousands of processors began to appear both in the United States and Japan, setting new computational performance records. By the end of the 20th century, massively parallel supercomputers were constructed with thousands of “off-the-shelf” processors. Significant progress in the first decade of the 21st century caused appearance of supercomputers with over 60,000 processors, reaching petaflop performance levels.

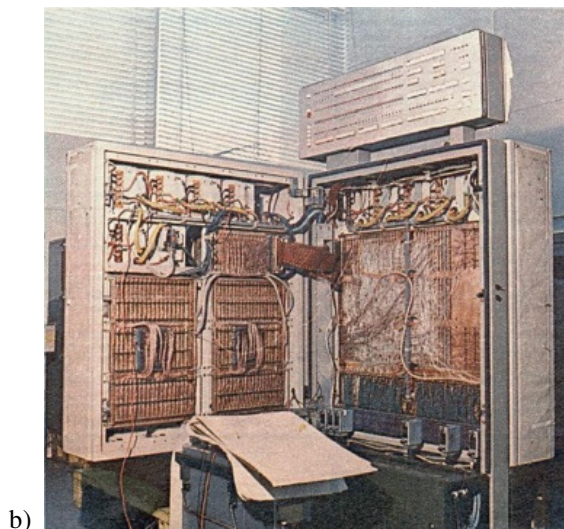


Figure 2.1.2. a) ATLAS supercomputer in National Museum of American History; b) the oldest Polish computer ODRA.

In the last years we have been observing amazing advance of computing power and capabilities of supercomputers. Over the course of 7 decades of development the

processors have moved from measuring thousands of operations per second to quadrillions of floating point operations per second. PC desktop from the late 1980s was about 10 times as fast as the on-board computer used on the way to the Moon by Apollo 11 astronauts. So, what then was considered to be a supercomputer, now could qualified as a satisfactory calculator.

Year	Power	Year	Power
1943	5 000 OPS	1994	170 GFLOPS
1944	100 000 OPS	1996	368 GFLOPS
1955	400 000 OPS	1997	1,34 TFLOPS
1960	1,2 MFLOPS	1999	2,38 TFLOPS
1964	3 MFLOPS	2000	7,23 TFLOPS
1969	36 MFLOPS	2002	35,86 TFLOPS
1974	100 MFLOPS	2004	70,7 TFLOPS
1976	250 MFLOPS	2005	280,6 TFLOPS
1981	400 MFLOPS	2007	478,2 TFLOPS
1983	941 MFLOPS	2008	1,1 PFLOPS
1984	2,4 GFLOPS	2009	1,76 PFLOPS
1985	3,9 GFLOPS	2010	2,5 PFLOPS
1989	10 GFLOPS	2011	10,5 PFLOPS
1990	23,2 GFLOPS	2012	17,59 PFLOPS
1993	143 GFLOPS	2014	33,86 PFLOPS

Table 2.1.1. Development of supercomputers power from the 40's to the present day, number of operations (OPS) and floating-point operations (FLOPS).

2.2. TOP 500 List

Since June of 1993, the TOP500 List has been presenting information on the world's 500 most powerful commercially available computer systems. It provides a resource for tracking and detecting supercomputing trends. Twice yearly TOP500, a new list of the 500 fastest computers is announced. It gives us the opportunity to observe just how rapidly the computer industry moves forward.

TOP500 currently relies on the Linpack benchmark – a specific set of criteria, which involves running a supercomputer through several dense linear equations and measuring its response times. That version of the benchmark allows the user to scale the size of the problem and to optimize the software in order to achieve the best performance for a given machine. This performance reflect the performance of a dedicated system for solving a dense system of linear equations. Since the problem is very regular, the performance achieved is quite high, and the performance numbers give

a good correction of peak performance. Although an alternative testing method is in the works.

2.2.1. Top 10 Supercomputers

Processing power of supercomputers is rated in FLOPS (short for floating-point operations per second). A floating-point operation is basically a computation using fractional numbers, so flops is a measurement of how many of these operations can be performed per second. Presently, performance of top supercomputers are measured in petaflops. The first system to break the 10-petaflop barrier was Japan's K Computer in early 2012. The major supercomputing locations are usually in the United States, Japan, Europe and China which are the leaders in this field (Fig.2.2.1).

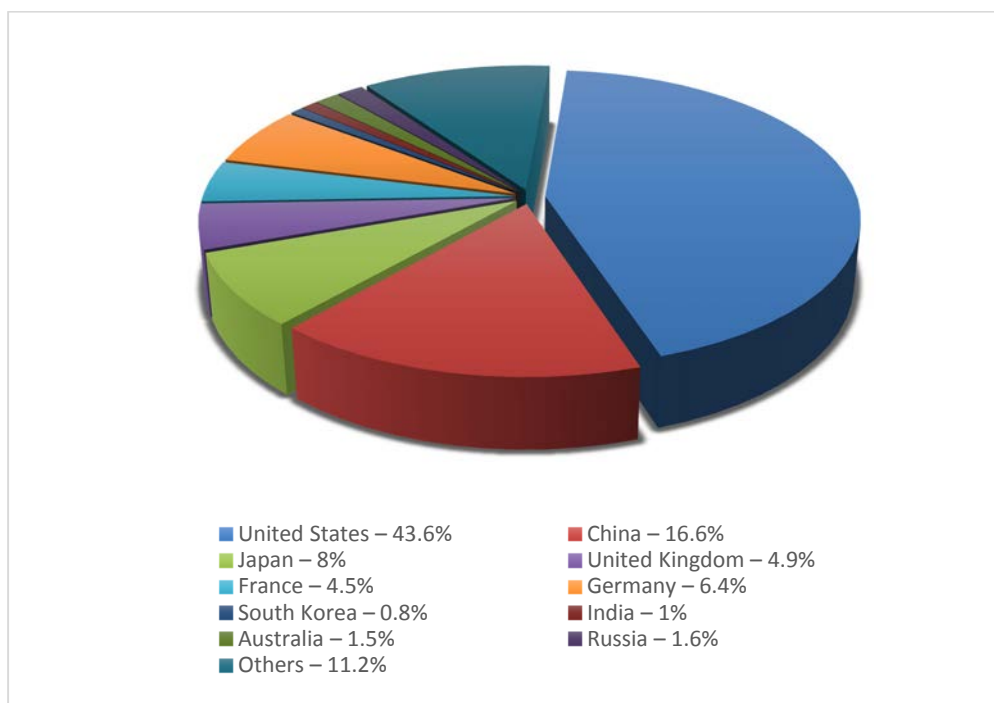


Figure 2.2.1. Pie chart showing share of supercomputers by countries from TOP500 supercomputers as of November 2014 [source: TOP500, Nov.2014].

Here are the champions at the end of 2014.

- **#10: Cray CS-Storm (United States)**

Site	Government
Manufacturer and vendor	Cray Inc.
Cores:	72,800
Rmax - Maximal LINPACK performance achieved	3,577 TFlop/s
Rpeak - Theoretical Peak Performance	6,131.84 TFlop/s
Nmax - Problem size for achieving Rmax	2,336,000
Power	1,498.90 kW
Memory	
Processor	Intel Xeon E5-2660v2 10C 2.2GHz
Interconnect	Infiniband FDR
Operating System	Linux

- **# 9: Vulcan (United States)**

Site	DOE/NNSA/LLNL
Manufacturer and vendor	IBM
Cores:	393,216
Rmax - Maximal LINPACK performance achieved	4,293.31 TFlop/s
Rpeak - Theoretical Peak Performance	5,033.16 TFlop/s
Nmax - Problem size for achieving Rmax	1,972.00 kW
Power	1,972.00 kW
Memory	393,216 GB
Processor	Power BQC 16C 1.6GHz
Interconnect	Custom Interconnect
Operating System	Linux

The Vulcan supercomputer is one of two in the top 10 operating at the U.S. Department of Energy's National Nuclear Security Administration (NNSA) at the Livermore National Laboratory (LLNL) in California. Vulcan is open up to U.S. businesses for collaborative work on projects that will increase scientific and technological advancement, improve U.S. competitiveness and enhance the high-performance computing (HPC) workforce, all in exchange for a share of the operational costs. Vulcan is also intended for collaboration with academic and research institutions in areas such as energy, security, atmospheric science and bioscience. Since the end of 2013 it has been placed on the 9th position.

- **# 8: JUQUEEN (Germany)**

Site	Forschungszentrum Juelich (FZJ)
Manufacturer and vendor	IBM
Cores:	458,752
Rmax - Maximal LINPACK performance achieved	5,008.86 TFlop/s
Rpeak - Theoretical Peak Performance	5,872.03 TFlop/s
Nmax - Problem size for achieving Rmax	
Power	2,301.00 kW
Memory	458,752 GB
Processor	Power BQC 16C 1.6GHz
Interconnect	Custom Interconnect
Operating System	Linux

While many computers on the list have been in function for the last couple years, JuQUEEN was built in 2012 as a replacement for another system, JUGENE, which was the ninth fastest system on the November 2010 TOP500.

Researchers at the Jülich-Aachen Research Alliance can submit proposals for projects to justify using some of JuQUEEN's cores, and if approved, they get some time with one of the most powerful supercomputers in the world. Time on this and two other powerful German supercomputers, can be requested through the Gauss Centre for Supercomputing (GCS) and the Partnership for Advance Computing in Europe (PRACE).

With that computing power, Jülich has become a computational and research simulation center for many disciplines: quantum systems, climate science, plasma physics, biology, molecular systems, and mathematical modeling and algorithms. Since the end of 2013 JuQUEEN has been placed on the 8th position.

- **# 7: Stampede (United States)**

Site	Texas Advanced Computing Center/ University of Texas
Manufacturer and vendor	Dell
Cores:	462,462
Rmax - Maximal LINPACK performance achieved	5,168.11 TFlop/s
Rpeak - Theoretical Peak Performance	8,520.11 TFlop/s
Nmax - Problem size for achieving Rmax	3,875,000
Power	4,510.00 kW
Memory	192,192 GB
Processor	Xeon E5-2680 8C 2.7GHz
Interconnect	Infiniband FDR
Operating System	Linux

Stampede, funded by the National Science Foundation Grant ACI-1134872 and built in partnership with Intel, Dell and Mellanox, went into production on January 7, 2013. Since that time, Stampede has been open for use by scientists and other researchers in all fields as part of the NSF's Extreme Science and Engineering Discovery Environment (XSEDE) program, a virtual system for openly sharing the computing power of 16 supercomputers and related resources. Ninety percent of the system is dedicated to XSEDE and the rest is at the discretion of the TACC's director. Qualified researchers at any U.S. institution can submit proposals via the XSEDE Web site. Since the end of 2013 it has been placed on the 7th position.

- **# 6: Piz Daint (Switzerland)**

Site	Swiss National Supercomputing Centre (CSCS)
Manufacturer and vendor	Cray Inc.
Cores:	115,984
Rmax - Maximal LINPACK performance achieved	6,271 TFlop/s
Rpeak - Theoretical Peak Performance	7,788.85 TFlop/s
Nmax - Problem size for achieving Rmax	4,128,768
Power	2,325.00 kW
Memory	
Processor	Xeon E5-2670 8C 2.6GHz
Interconnect	Aries interconnect
Operating System	Cray Linux Environment

Piz Daint resides at the Swiss National Supercomputing Centre (CSCS) and will be put to use modeling weather and climate patterns and performing scientific computation in a variety of other fields.

The Piz Daint is also one of the most energy efficient supercomputers, with an efficiency level of 3,185.9 megaflops per watt (MFLOPS/W) [sources: Green500]. It's the only supercomputer to make the top 10 both in the TOP500 and GREEN500 lists. GREEN500 takes all the supercomputers in the TOP500 list and ranks them by energy efficiency. Piz Daint's hybrid architecture, which uses both traditional CPUs (central processing units) and more energy-efficient GPUs, helps keep its energy usage low. Since the end of 2013 it has been placed on the 6th position.

- **# 5: Mira (United States)**

Site	DOE/SC/Argonne National Laboratory
Manufacturer and vendor	IBM
Cores:	786,432
Rmax - Maximal LINPACK performance achieved	8,586.61 TFlop/s
Rpeak - Theoretical Peak Performance	10,066.3 TFlop/s
Nmax - Problem size for achieving Rmax	
Power	3,945.00 kW
Memory	
Processor	Power BQC 16C 1.6GHz
Interconnect	Custom Interconnect
Operating System	Linux

The supercomputer was constructed by IBM for Argonne National Laboratory's Argonne Leadership Computing Facility with the support of the United States Department of Energy, and partially funded by the National Science Foundation. Mira will be used for scientific research, including studies in the fields of material science, bio-sciences climatology, transportation seismology, and computational chemistry.

Researchers who submit proposals for the Innovative and Novel Computational Impact on Theory and Experiment program (INCITE) through the U.S. Department of Energy's Office of Science can claim processor time on Mira. Argonne's computing program is open to so many researchers outside of the DOE (United States Department of Energy) who can apply for time on Argonne's system, and they're awarded a set number of hours on the system. Sixty percent of the computer's capacity goes to their research, while 30 percent goes towards the Advanced Science Computing Research Leadership Computing Challenge. The final 10 percent is reserved for urgent, time-sensitive computations. Since the end of 2013 it has been placed on the 5th position.

- **# 4: K computer (Japan)**

Site	RIKEN Advanced Institute for Computational Science (AICS)
Manufacturer and vendor	Fujitsu
Cores:	705,024
Rmax - Maximal LINPACK performance achieved	10,510 TFlop/s
Rpeak - Theoretical Peak Performance	11,280.4 TFlop/s
Nmax - Problem size for achieving Rmax	11,870,208
Power	12,659.89 kW
Memory	1,410,048 GB
Processor	SPARC64 VIIIfx 8C 2GHz
Interconnect	Custom Interconnect
Operating System	Linux

Fujitsu's K computer, named for the Japanese word "kei" (京), meaning 10 quadrillion (10^{16}), is the only supercomputer in Japan that made the top 10. It was the fastest supercomputer in the world on both 2011 lists, but has since been placed on the 4th position. It had just become the first computer to surpass the 10-petaflop barrier.

The K computer is located at Japan's RIKEN Advanced Institute for Computational Science, where it performs scientific operations including global disaster prevention, meteorology and medical research. Riken's research teams use computer simulation to study and predict biological processes in the human body at the molecular, genetic, cellular and organ levels. For example, simulation of the human heart which had taken two years, for instance, is accomplished in just one day. Obtaining heart simulations in a short time is expected to contribute to the field of cardiology by gaining a more accurate understanding of heart disease mechanisms. Also, K computer can provide a breakthrough in the area of drug discovery.

Unlike many of the other supercomputers on the list, it doesn't run on IBM architecture. The K computer uses Fujitsu's own SPARC64 VIIIfx octo-core processors. Its 705,000 computer cores help it churn through operations at an incredible pace.

K computer is being used in a broad range of fields including drug discovery, earthquake/tsunami research, weather forecasting, space science, manufacturing and material development. K computer is openly available for use by researchers throughout the world. Many scientists and technical researchers from institutes, universities and industry are currently using the K computer. Software is continually being developed and ported to K. The supercomputer is becoming useful in even more areas. K's fast calculation speed allows high-resolution simulations which had not been possible before, leading to new discoveries and the opening of new research fields.

- **# 3: Sequoia (United States)**

Site	DOE/NNSA/LLNL
Manufacturer and vendor	IBM
Cores:	1,572,864
Rmax - Maximal LINPACK performance achieved	17,173.2 TFlop/s
Rpeak - Theoretical Peak Performance	20,132.7 TFlop/s
Nmax - Problem size for achieving Rmax	
Power	7,890.00 kW
Memory	1,572,864 GB
Processor	Power BQC 16C 1.6GHz
Interconnect	Custom Interconnect
Operating System	Linux

Sequoia was the top ranked supercomputer on the June 2012 TOP500 list, but dropped to No. 2 in November 2012, and now rests at No. 3. It is a petascale Blue Gene/Q supercomputer constructed by IBM for the National Nuclear Security Administration as part of the Advanced Simulation and Computing Program (ASC).

In January 2013, the Sequoia sets the record for the first supercomputer using more than one million computing cores at a time for a single application. The Stanford Engineering Center for Turbulence Research (CTR) used it to solve a complex fluid dynamics problem – the prediction of noise generated by a supersonic jet engine and also simulating nuclear explosions. Sequoia is also used for nuclear weapon security and to make large-scale molecular dynamics calculations. Since 2013 it has been placed on the 3rd position.

2: Titan (United States)

Site	DOE/SC/Oak Ridge National Laboratory
Manufacturer and vendor	Cray Inc.
Cores:	560,640
Rmax - Maximal LINPACK performance achieved	17,590 TFlop/s
Rpeak - Theoretical Peak Performance	27,112.5 TFlop/s
Nmax - Problem size for achieving Rmax	
Power	8,209.00 kW
Memory	710,144 GB
Processor	Opteron 6274 16C 2.2GHz
Interconnect	Cray Gemini interconnect
Operating System	Cray Linux Environment

Titan is located at the Oak Ridge National Laboratory (ORNL) run by the Oak Ridge Leadership Computing Facility (OLCF). Titan nabbed the No. 1 spot on the Top500 the month after it was deployed in late 2012, but dropped down to No. 2 in the June 2013 rankings, where it remains.

Titan is one of two supercomputers in the top 10 that incorporates NVIDIA GPUs to increase performance while keeping the power consumption lower than it would be for a similarly powerful all-CPU system. Utilizing the GPUs to run applications at the speed of which the system is capable requires a new approach to programming its software, however. To address this, the OLCF has partnered with Cray and NVIDIA to create the Center for Accelerated Application Readiness (CAAR), which is hard at work coming up with best practices for code development.

The Titan uses a combination of CPUs by AMD and GPUs by NVIDIA to conduct research for the U.S. Department of Energy, including studying the effect of extending the life cycles of nuclear power plants, the viability of new biofuels, population dynamics, development of solar technology and climate change modeling. Titan is available for any scientific purpose; access depends on the importance of the project and its potential to exploit the hybrid architecture. As with Mira, researchers can submit proposals through the U.S. Department of Energy's INCITE.

1: Tianhe-2 (MilkyWay-2) (China)

Site	National Super Computer Center in Guangzhou
Manufacturer and vendor	NUDT
Cores:	3,120,000
Rmax - Maximal LINPACK performance achieved	33,862.7 TFlop/s
Rpeak - Theoretical Peak Performance	54,902.4 TFlop/s
Nmax - Problem size for achieving Rmax	9,960,000
Power	17,808.00 kW
Memory	1,024,000 GB
Processor	Intel Xeon E5-2692v2 12C 2.2GHz
Interconnect	TH Express-2
Operating System	Kylin Linux

The Tianhe-2 (Tianhe-2 or TH-2 (Chinese: 天河-2; pinyin: tiānhé-èr; literally: “Heavenriver-2”, that is, “Milky Way 2”) and since 2013 it has been placed on the 1st position.

The development of Tianhe-2 was sponsored by the 863 High Technology Program, initiated by the Chinese government, the government of Guangdong province, and the government of Guangzhou city. It was built by China's National University of Defense Technology (NUDT) in collaboration with the Chinese IT firm Inspur.

TH-2 is used for education and research. Tianhe-2 runs on a custom version of the Ubuntu Linux operating system called Kylin (freely available, open-source operating system tailored specifically for Chinese users), which was developed through a partnership between the NUDT, the China Software and Integrated Circuit Promotions Centre (CSIP) and Canonical (creators of Ubuntu).

2.3. Supercomputers in Poland

Today, it is very difficult to point out any field of science or technology that would not use supercomputers. In Poland, as in other countries, the main beneficiaries of the supercomputer power are particle physics, quantum chemistry and molecular biology as well as materials science, medicine and finally numerical weather forecast. These machines are used by scientists involved in national and international research projects, for example the recent discovery of the Higgs boson was made in cooperation with Polish scientists using Polish supercomputers.

So far, the fastest supercomputers in Poland have been installed mainly in university computing centers, such as Gdansk TASK (supercomputer Galera), Poznan Supercomputing and Networking Center in Institute of Bioorganic Chemistry and Cracow Cyfronet (supercomputer Zeus). But, it also happened that some machines were introduced on TOP500 list Polish by private companies, e.g., Cluster Platform of Allegro Group, the popular auction portal.

There are several computing centers in Poland which offer their computing power for the research purposes:

- Wroclaw Center for Networking and Supercomputing (WCSS)

One of WCSS main activities is to provide computing clusters for scientific research. These clusters are referred to as High Performance Computers (HPC). The use of WCSS High Performance Computers is free of charge and is done on the basis of so called computing grants. The grants applications are accepted on a prescribed basis. The parameters of the computing infrastructure with the data storage system are:

system performance	72,5 TFlops
operating memory	14,5 TB
disk capacity	895 TB
mass storage	2,82 PB

- Poznan Supercomputing and Networking Center (PCSS)

High Performance Computing centre within PSNC: provides computing power, disc space and archiving systems for science, business and public institutions. It appeared on the TOP500 list of the most powerful computing

systems in the world. The computing capacity includes systems with distributed and shared memory of varied architectures (parallel vector, multi-processor SMPs and clusters) connected with fast local networks (InfiniBand, Gigabit Ethernet and FastEthernet).

system performance	310 TFlops
operating memory	156 TB
disk capacity	153 TB
mass storage	672 TB

- Academic Computer Centre CYFRONET AGH, Cracow

Academic Computer Centre CYFRONET AGH was established by the National Committee for Scientific Research as the unit leader in the operation and expansion of the hardware base of the high performance computers (HPC). CYFRONET provides the scientific community with computing power as well as disk and tape storage resources.

Zeus Group:	Zeus	Zeus vSMP	Zeus GPGPU	Zeus BigMem
system performance	169 TFlops	8 TFlops	136 TFlops	61 TFlops
operating memory	23 TB	6 TB	3,6 TB	26 TB
cores	17516	768	528	6656

- Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw

Centre participates in many projects and initiatives, national and international, in which its competence and computing resources are provided for the scientific community creating also the conditions for participation in projects of international scope. Centre offers powerful computers – including several thousand processors in the AMD machines, Intel, IBM BlueGene (see BlueGene on following pages), Power7. The own specialized softwares are installed on the machines computing center or supported through national licenses (*more information: <http://www.icm.edu.pl>*).

- Academic Computer Centre in Gdansk (CI TASK)

The Supercomputers Group maintains and manages all modern computers (IBM, SGI, SUN, clusters) installed in CI TASK. They are equipped with a high-capacity archiving system (ATL, HP, and EXABYTE). A very rich software installed on computers of the Centre (34 professional packages)

makes it possible to perform large scale calculations in various fields of science and technology. The group also delivers expert advice in software and servers security. Its latest achievement is the SGI cluster project that resulted in effective supercomputer workload.

Computer:	Galera
system performance	50 TFlops
operating memory	10752 GB
disk capacity	107,5 TB
cores	5376

2.3.1. Polish TOP500

Supercomputers of the TOP 500 are significantly different from each other and their performance is not always the most significant value. Among them, there are ones supporting the accelerators, such as NVIDIA Tesla PHI Xeon or AMD Fire-Stream. Their architecture – despite of enormous theoretical computing power – has a number of hardware limitations and very narrow range of applications.

In Poland, investments in infrastructure are necessary to keep pace with the global economy. The computing power is developed in supercomputing centres to assure the stable development of research groups benefiting from the supercomputers.

Two Polish supercomputers were on the list of 500 fastest in the world released in November 2014. The best one in Poland is Zeus cluster of Academic Computer Center Cyfronet at AGH University of Science and Technology. It took 211th place. Second Polish supercomputer on the is BlueGene/Q from Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw. It took 342th place.

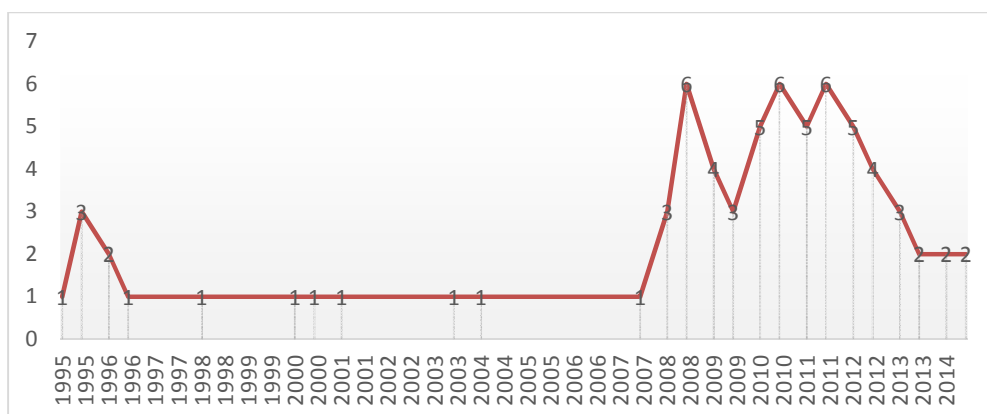


Figure 2.3.1. Presence of Polish supercomputers on TOP500 list.

1 in Poland: Zeus (Cracow), ranked 81 in 2011, 211 in 2014.

Site	Cyfronet
Manufacturer and vendor	Hewlett-Packard
Cores:	25,468
Rmax - Maximal LINPACK performance achieved	266.942 TFlop/s
Rpeak - Theoretical Peak Performance	373.89 TFlop/s
Nmax - Problem size for achieving Rmax	
Power	
Memory	60 TB
Processor	Xeon X5650 6C 2.66GHz
Interconnect	Infiniband QDR
Operating System	Linux

Cluster Zeus is composed of more than one thousand three hundred individual blade Hewlett-Packard servers, interconnected by Infiniband network of speeds 40 Gb/s. Some of the servers are equipped with GPGPU NVIDIA cards, what allows faster computation in case of selected scientific algorithms. The Zeus is working under supervision of Scientific Linux operating system – an European-American GNU/Linux distribution, based on Red Hat Linux and aimed at scientific applications. The new devices use technology of cooling microprocessors with liquid directly on supercomputer boards.

According to Cyfronet estimates, computing power of Zeus allows it to perform more than 370 trillion mathematical operations per second what corresponds to the power of more than 20 thousand standard PCs. Number of computational tasks performed by Zeus per year exceeded 8,000,000.

Scientists from all over Poland can use its enormous power to solve increasingly complex computational problems. Supercomputer Zeus is used for example for modeling energy projects, calculations in the development of high energy physics (e.g. in the work for CERN) and the complex calculations in chemistry, biology and nanotechnology. It also helps in research on the properties of the materials necessary for the production of fuel cells, analgesics, on effects of gamma radiation used in cancer therapy in order to minimize the intensity of radiation from the simultaneous maximization of the effect of treatment, and many others. Access to resources is free of charge for all persons conducting scientific activity.

Supercomputer Zeus is a part of a national grid infrastructure created within the PL-Grid project (co-funded by the European Regional Development Fund as part of the Innovative Economy program).

2 in Poland: BlueGene/Q Nostromo (Warsaw), ranked 143 in 2011, 342 in 2014.

Site	ICM, University of Warsaw
Manufacturer and vendor	IBM
Cores:	16,384
Rmax - Maximal LINPACK performance achieved	188.967 TFlop/s
Rpeak - Theoretical Peak Performance	209.715 TFlop/s
Nmax - Problem size for achieving Rmax	1,409,023
Power	82.19 kW
Memory	16 TB
Processor	Power BQC 16C 1.6GHz
Interconnect	Custom Interconnect
Operating System	Linux

Nostromo – a supercomputer which operates in Interdisciplinary Centre for Mathematical and Computational Modelling at University of Warsaw – was ranked on 342th place among other TOP500 machines worldwide. The system consists of a processing unit of IBM's Blue Gene / Q equipped with 1024 compute nodes.

Currently Nostromo is the most energy-efficient computer in Poland, ranked at 28th position according to the Green500 list released on November 2014 (for comparison Tianhe-2 took 64th place Sequoia – 48th and Zeus– 440th) with a score of ca 2,3 GFlops per watt power consumption.

It's one of Europe's largest computing environments for research in biomedicine and biotechnology eg. to carry out large-scale simulation in biomolecular modelling and neurobiology.

Nostromo system was purchased and installed within the Project Center for Pre-clinical Research and Technology (CePT) which is the largest biomedical and biotechnological project in the Central and Eastern Europe. The aim of the project is to create a Warsaw scientific center consisting of closely cooperating environmental research centers, in which the research would be focused on the most common lifestyle diseases, in particular cancers, neurological, cardiovascular diseases associated with aging. CePT is also co-funded by the European Regional Development Fund as part of the Innovative Economy program.

CIS Supercomputer (Swierk)

Site	Swierk Computing Center (CIS), National Center for Nuclear Research
Manufacturer and vendor	HP
Cores:	ultimately up to 20,000
Rmax - Maximal LINPACK performance achieved	
Rpeak - Theoretical Peak Performance	ultimately up to 500 TFlop/s
Power	
Memory	ultimately up to 130 TB
Processor	1800 Intel Xeon E5-2680v2 oraz 120 AMD Opteron 6276
Interconnect	Infiniband QDR
Operating System	Cray Linux Environment

Supercomputer from NCBJ officially started on November 13, 2014 and became one of the largest computer clusters in Poland. Its performance up to 500 TFlop/s would be enough to put CIS' among the top 100 supercomputers in the world, according to the TOP500 list.

Similar like majority of modern nuclear reactors the supercomputer's hardware is cooled down with hot water. That innovative approach is unique not only in Poland, but also in Europe and it allows to maximize the use of available computational resources, and simultaneously to significantly lower supercomputer operational costs. This hot-water-based solutions turned out to be extremely efficient: using just 12–20 kW electric power. It is capable to dissipate 600 kW thermal power and because of it that technique is recommended by European Commission as the target solution for any future large computational facility.

It's no coincidence that the facility is based next to the National Centre for Nuclear Research. The main target of the CIS project is to support the Polish energy sector, specifically for nuclear power in line with the Polish Nuclear Power Programme. Supercomputer applications in that field include design/optimization calculations for power generation/distribution facilities/networks, nuclear reactor safety analyses (for example, safety analyzes and models of various types of reactors), radiation protection monitoring/threat simulations, support for crisis management decision-makers. An important area of application of this supercomputer will be also research and development in the areas associated with nuclear and conventional energetics, as well as other research on complex systems, physics of fundamental interactions and astrophysics. It also helps in the processing of data from experiments performed at the famous Large Hadron Collider near Geneva.

The project is co-funded from European Regional Development Fund within framework of the Innovative Economy Operational Programme (over 83 million

PLN) and a target subsidy from Ministry of Science and Higher Education (over 14.5 million PLN).

CIS supercomputer is available for scientific purpose. Researchers can submit proposals through the Swierk Computing Center (templates of applications are available on the www.cis.gov.pl) and after positive verification of such an application the access to resources is granted.

2.3.2. New Polish investments

Gdańsk University of Technology is planning to build up a supercomputer worth PLN 30 million by the end of 2014, which will be the first machine in Poland capable of performing a quadrillion mathematical operations per second.

This supercomputer will be the central part of the project Centre of Excellence for Scientific Application Manufacturing Infrastructure at Gdańsk University of Technology. It will be used in many fields – from physics and chemistry to medicine, for example for the analysis of sound recordings and images. In endoscopy the computer will analyze the recordings from gastrointestinal tract examinations and identify images that suggest the occurrence of adverse changes in the body. Other new technologies can be in the field of energy (shale gas), the environment (level and water pollution).

Part of the costs will be covered from European Regional Development Fund within framework of the Innovative Economy Operational Programme.

These ambitious plans is already being followed by another supercomputing project named Prometheus, from the AGH University of Science and Technology in Cracow. It should be completed in the first decade of January 2015. Its theoretical performance will reach nearly 1.7 petaflops (1 PFLOPS is 10 to the power of 15 operations per second) – four times more than Zeus, the current supercomputer utilized by the Cracow University. According to TOP500 list, the new computer will be among the 30 fastest in the world and the 12 fastest in Europe.

This new supercomputer will represent the world's largest installation of HP Apollo 8000 servers connected by lighting – fast Infiniband broadband, capable of running at speeds of 56 Gb/s. It will be connected to two file systems with a total capacity of 10 PB and 150 GB/s bandwidth. The computer's efficiency will be ensured by 41,000 Intel Haswell-EP CPUs and 215 DDR4 TB of working memory.

Prometheus will be also one of the most energy-efficient computers in its class, because of its innovative liquid-cooled design. It will consume approximately 650 kW, which is only about 25 percent more than Zeus.

Prometheus will be used for research from such areas as chemistry, physics, astrophysics, biology, power industry, technology, nanotechnology and medicine. As in

the case of Zeus, scientists will have the access to the Prometheus resources free of charge.

Project costs are fully funded from European Regional Development Fund within framework of the Innovative Economy Operational Programme.

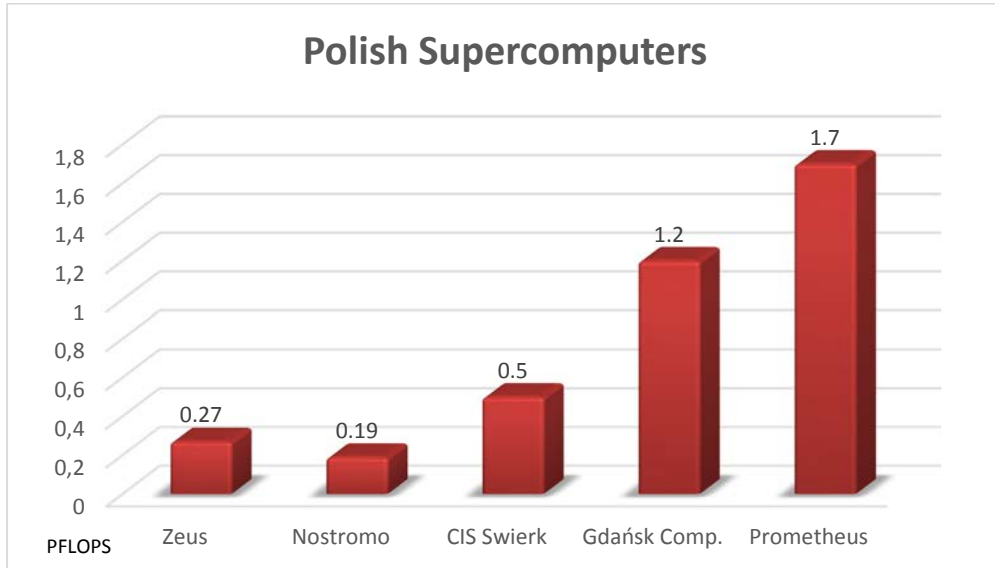


Figure 2.3.2. Growth of computing power of Polish supercomputers.

2.4. Future of supercomputers

In 2008, the IBM Roadrunner broke the one-petaflop barrier: one quadrillion operations per second. Cray began work on a successor to the supercomputer Titan XC30 machine called Cascade, which achieved a performance of 100 PFLOPS. IBM announced that their architecture Blue Gene / Q achieves 100 PFLOPS.

Another milestone is exaflop computer – being currently at development. It should be constructed by 2019 – performing at 10^{18} FLOPS, i.e. a thousand times faster than the petaflop computers.

According to Intel Corporation, the current growth rate of the computing speed will be maintained until at least 2029 years, when the rise of computers performance would achieve zettaflops (1021 tryliard FLOPS).

Supercomputers are rare and expensive resources that should be used not to accommodate the largest number of users or the highest computing power, but to be able to effectively solve the largest, most difficult, and most important scientific problems.

2.5. Short history of IPPT computational resources

The first “supercomputer” Odra 1204 was installed at the Institute of Fundamental Technological Research of the Polish Academy of Sciences in 1972. The implementation of the new methodology of doing Science, on the basis of numerical simulations, was introduced with great passion by Prof. Jacek Mączyński, head of the Numerical Calculations Laboratory at the Institute. Since 1984 computer Odra became technologically obsolete and was gradually replaced by semi-legally imported (because of embargo) Minicomputers ZX Spectrum and Commodore 64, which had found many applications in numerical computation or to control experiments. At the time in the group of Prof. Bogusław Radziszewski the new operating system Warsaw Basic was developed at IPPT, significantly extending the computational capabilities of C64.

In 1992 another computational system based on Sun Cluster, theoretically allowing multiprocessor calculations was created. However, imperfect system of Ethernet quickly discouraged prospective users and the system has been superseded by individual IBM compatible PC computers. In 2005 on the basis of these resources, a multiprocessor cluster-based operating system OpenMosix was built, which until 2010 has successfully enabled the realization of a relatively complex calculations in fluid mechanics [more info: <http://fluid.ippt.gov.pl/mosix/TheopenMosix.pdf>]. Recent important step towards digitization of IPPT is launched in 2013 by instalment of the Grafen cluster. We do hope it wouldn't be the last investment, as the progress in the computer industry is astonishing fast.

Prepared on the basis of the materials available on the Internet.

Grid computing

Rafał Gawlik

3.1. Differences between grid and cluster

Grid computing is the collection of computer resources from multiple locations connected together to provide computing service. Grid can be thought of as a powerful virtual computer, created by connection of many systems sharing different resources.

Concepts of grid and cluster are often confused. In both cases, we are dealing with systems consisting of multiple computers. What distinguishes the grid computing from the cluster type system is that the grid is more loosely coupled, heterogeneous, and geographically dispersed. Whereas the cluster usually consists of servers located in one place and supported by a single operating system.

Characteristic of the cluster system

- strongly integrated,
- single operating system on all machines, single operating system image,
- centralized management and shared job queue,
- basically designed to provide one type of service.

Characteristic of the grid system

- weakly integrated,
- consists of different machines and system controlled by different operating systems,
- distributed management,
- usually covers a wide range of services.

A single computer in the cluster system is called node. There are compute nodes or service nodes which do not affect the cluster efficiency, but they provide services necessary to cluster's proper operation, e.g., managing services, disc services, access services.

The origins of grid systems are big networks consisting of thousands of PCs. One of the first grid system was SETI@home, which allowed each Internet user to participate in the search of extraterrestrial civilizations signals, using their PCs.

The grid is the form of distributed computer system, whereas – in some cases – grid processing might be treated as parallel processing based on complete computers connected to network by conventional link e.g. Ethernet type. That distinguishes grid from supercomputer, which consists of many cores or processors connected by dedicated large capacity data bus.

The size of the grid varies a considerable amount – from small ones, limited to local company or institution network to big, public grids, based on cooperation of many institutions and computer networks. Grid technologies are used to solve scientific problems, which is often supported by volunteers connected to the grid. They are also used in a commercial way such as new medicine search, economic forecasts, seismic analyses and e-commerce backoffice processing or web services. Nowadays, commercial services of that kind are becoming more popular as well as grids consisting of computer clusters.

3.2. Examples of grid systems in Poland

- **Polish Grid Infrastructure** (www.plgrid.pl)

PL-Grid is the IT platform built between 2009–2012. It is based on computer clusters, serving e-Science in different fields. Infrastructure supports scientific research by integration of experimental data and results of advanced computer simulation conducted by geographically dispersed teams. Users can use specialist packages from the variety of scientific fields: biology, quantum chemistry, physics, numerical calculations, simulation.

- **HPC Infrastructure for Grand Challenges of Science and Engineering (POWIEW)** (www.wielkiewyzwania.pl)

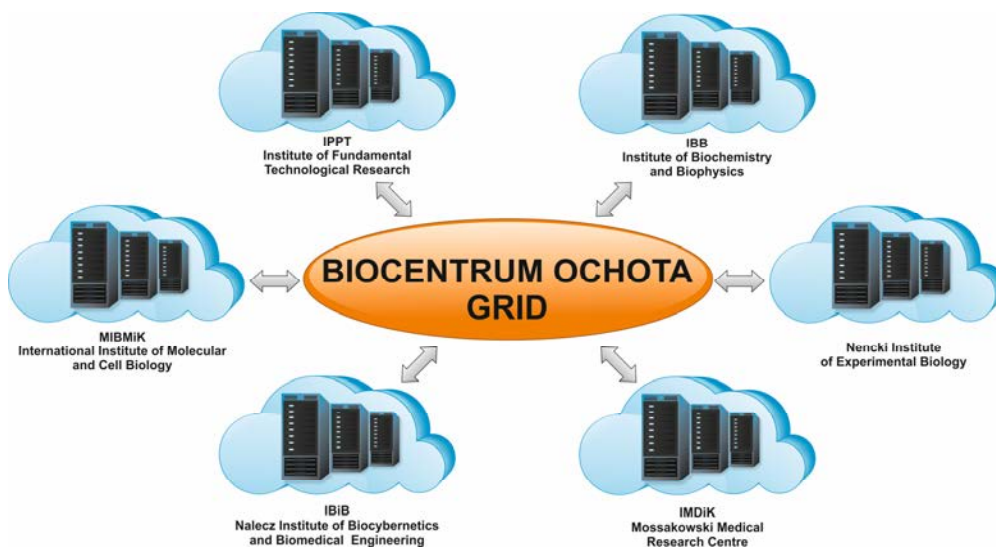
Main objective of the program is designing and providing computing infrastructure with the scientific and programming support, creating national competence centre in the field of developing computational models for leading Peta-scale generation supercomputer architectures and creating competence centre in the field of multiscale data analysis and visual modeling. The systems available in the project: massively parallel processing systems, fat-node multiprocessing systems, symmetric multiprocessing systems, accelerated heterogeneous systems.

- **National CLUSTER of Linux Systems (CLUSTERIX)**

The Project was finished in 2005. The objective of the project was to provide academic users with sources, services and applications in order to carry out scientific and research work that were requiring extremely high performance computing. Among other things Clusterix was used to modeling of thermomechanical phenomena in solidifying castings, large-scale simulation of blood flow in the micro-capillaries, designing of molecular electronic systems and simulation of mechanosynthesis.

3.3. Grid BIOCENTRUM OCHOTA

Grid Biocentre was created as a compromise between financial abilities and computing performance. Grid consists of six clusters of the following institutes:



The total equipment resources are:

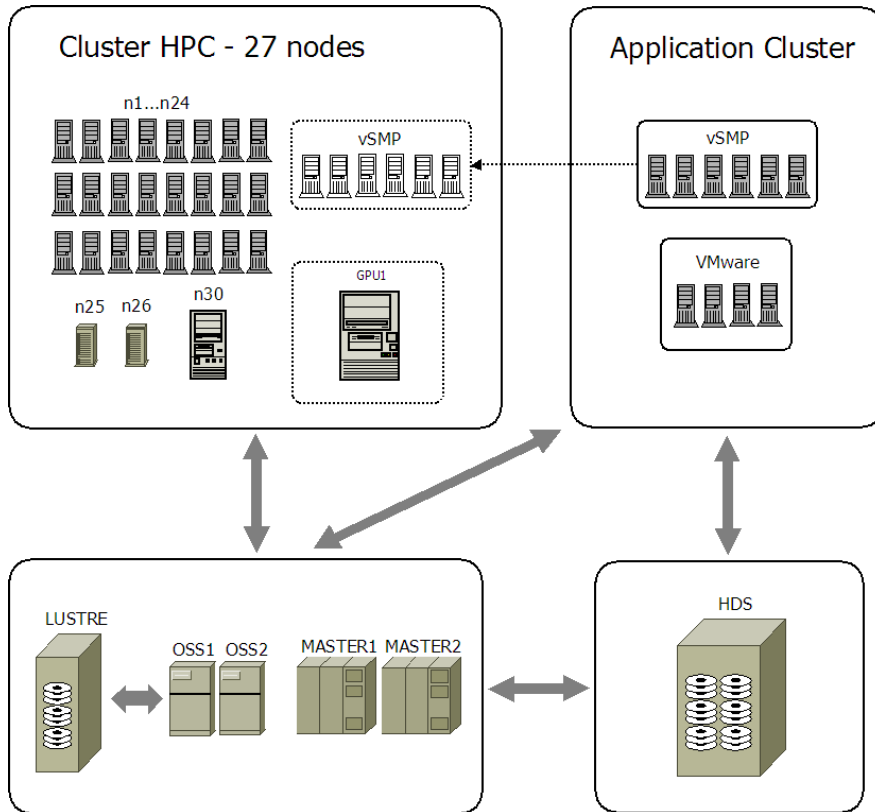
- 4000 cores,
- 10 TB RAM,
- 40 GB/s InfiniBand links,
- 42 TFLOPs.

3.4. Grafen cluster

One of system elements – Grafen cluster is located at the Institute of Fundamental Technological Research.

3.4.1. Equipment

The diagram below shows the cluster structure.



HPC

HPC (High Performance Computing) part is designed to distributed computing. It consists of 27 machines:

- 24 compute servers:
 - Xeon X5650 2,66 GHz six core processors,
 - 24 GB RAM memory,
- 2 compute servers:
 - 2 E5-2670 2,66 GHz, eight core processors,
 - 32 GB RAM,
- 1 compute server:
 - Xeon X7460 2,66 GHz six core processors,
 - 132 GB RAM,
- Inter nodes connections with QDR InfiniBand 40 Gb/s network.

HPC provides 344 physical cores and 772 GB RAM. The system has no physically shared memory so in order to take full advantage of its capability, software enabling

sharing the memory on separate machines, need to be installed, e.g., MPI (Message Passing Interface).

Application cluster

Application cluster consists of vSMP part and a part operating virtual machines by VMware software. Available virtual machines are for example:

- Windows 2008 server with graphic and scientific applications,
- Mail Server gateway,
- MATLAB compute Server,
- License Server.

vSMP part consists of six compute servers, each server contains:

- Xeon X5650 2,66 GHz six core processors,
- 96 GB RAM memory DDR3 1333 MHz,
- connections between nodes by QDR Infini Band 40 Gb/s network.

From the user's point of view vSMP is seen as a single 72 computing core computer with 576 GB RAM memory, available under the name of *vsmp1* in HCP cluster.

GPU

GPU is an efficient computer used to carry out calculation on graphic processors. Equipment configuration is shown below:

- Actina Solar G 200 s4+ server,
- 4 Tesla C2050 cards with 1150 MHz clock and 10,5 GB memory GDDR5.

The set provides 4 TFLOPS compute performance for fixed-point computing and 2 TFLOPS for floating point computing. GPU is available as *gpu1* node in HPC cluster.

Service nodes

- MASTER access nodes where headnode virtual machine is installed to serve cluster log in,
- OSS service nodes support LUSTRE file system where users' data is stored (LUSTRE – cluster file system on Linux platform),
- disk array HDS (Hitachi Data Storage).

Network connections

Parts of cluster and nodes are connected by high bandwidth links:

- InfiniBand FullCBB QDR 40 Gbit/s Network, providing fast communications between HPC compute nodes,
- Ethernet 10 Gb/s network providing communications between other parts of cluster,
- Ethernet 10 Gb/s for other elements of GRID Biocentre.

3.4.2. Software

Grafen cluster operates under control of Scientific Linux operating system. Written in C, Fortran, Python languages software needed to start computing programs and scripts to start MATLAB, COMSOL, LS DYNA and others are also installed there.

The list of installed software can be revealed by typing command `rpm -qa` on an access node or any compute node. The same packages are installed on all nodes.

Task Manager

Task Manager is the special software, installed on the cluster. It manages tasks, performing the following services:

- Informs users about available resources and the level of their exploitation,
- Accepts tasks to be performed and runs them at the most suitable moment,
- Takes into the account the priorities of tasks,
- Allows monitoring of task process,
- Builds task queues.

Task manager often has integrated functions of job scheduling. It tries to perform above mentioned tasks optimally i.e. the hardware resources should be used as much as possible, and the tasks should end as soon as possible. Unfortunately, to achieve that the task manager must postpone some tasks, and that is why the task is not always started at the moment of entering the command as on ordinary personal computer. Additionally, the task manager may take into account some additional criteria (e.g. to achieve certain predefined ratio between the resources allocated to the tasks of individual, organizational units, it may reward tasks meeting some specific criteria by fast execution etc.). The function of the task manager in IPPT cluster is performed by *TORQUE* together with *Maui*, both maintained as open source by Adaptive Computing Enterprises, Inc.

TORQUE manager is a fork of OpenPBS – open source Portable Batch System. Its primary task is to allocate computational tasks, i.e. batch jobs, among the available computing resources. Maui Cluster Scheduler is designed for use on clusters and supercomputers. It is capable of supporting multiple scheduling policies, dynamic priorities, reservations, and fairshare capabilities. Maui improves the manageability and efficiency of the cluster.

Another job scheduler is LoadLeveler developed by IBM. It is a parallel scheduling system that matches each job's processing needs and priority with available resources and special instructions for maximum resource utilization. LoadLeveler was primarily available only for AIX operating system; however, now, it is available for POWER and x86 architecture Linux systems.

Third widely used job scheduler is SLURM, that stands for Simple Linux Utility for Resource Management. It is a free open source job scheduler for the Linux kernel used by many world's clusters and supercomputers including Tianhe-2 that is the world's fastest computer (by Top500: <http://www.top500.org/>, as of November 2014). SLURM was originally developed at Lawrence Livermore National Laboratory and now is the default workload manager on LLNL systems including the Sequoia supercomputer, which was the number one of the Top500 by June 2012. More information can be found on the project page: <http://slurm.net/community/>.

Compilers

Three packages of C, C++ and Fortran languages compilers are installed on nodes:

- gcc – GNU Compiler Collection,
- intel – Intel Compilers,
- pgi – PGI compilers.

GCC is available in standard system paths, the remaining two sets can be found in directory */opt*.

MPI Libraries

Message Passing Interface (MPI) is a communication protocol which is a standardized message-passing system between processes of parallel programs working on one or more computers. It enables data transfer between particular program processes carried out on processors of machines that are nodes of the cluster. MPI libraries allow to write parallel programs and starting the software using parallel processing on the cluster.

The following MPI libraries have been installed on Grafen cluster:

- Intel MPI supplied with Intel Cluster Studio,

- MVAPICH highly compatible with MPICH (available both in version 1 and 2), but optimized to work with Infini Band network,
- Open MPI.

All above mentioned libraries (except Intel MPI) are available both in compiled version with compiler Intel and GCC. To check which MPI library is really active which *mpirun* command might be used.

Environment Modules

The Environment Modules system is a tool to help users to manage their Linux shell environment, by allowing groups of related environment-variable settings to be made or removed dynamically. The modules system is based on *modulefiles*, which specify groups of environment settings that need to be made together. Environment Modules modulefiles are written in the *TCL* (Tool Command Language) and are interpreted by the *modulecmd* program via the module user interface. Environment Modules modulefiles can be loaded, unloaded, or switched on-the-fly while the user is working and can be used to implement site policies regarding the access and use of applications. To check what modules are available command *module avail* must be used.

Others

Aforementioned Intel and PGI compilers are only the part of installed software packages, respectively Intel Cluster Studio XE and PGI Workstation create user friendly environment. In addition to compilers they also include tools for optimizing and debugging programs, particularly supporting parallel processing, as well as extensive documentation. Detailed information might be found on relevant websites.

On *gpu1* node, in */opt/cuda* directory, *CUDA* software and *SDK* (Software Development Kit) are installed in */opt/NVIDIA_GPU_Computing_SDK* directory. *CUDA* (Compute Unified Device Architecture), created by NVIDIA, is a universal multi-core architecture (mainly graphic cards) enabling the use of multi-cores computing power to solve general numerical tasks more efficiently than in traditional general-purpose sequential processors.

Scientific software

Rafał Gawlik

Apart above mentioned libraries and compilers, scientific programs used to perform calculations or pre/post-processing have been installed on the cluster. Most of them installed on standalone computer use graphic user interface (GUI), but preferable method to work on cluster is batch mode. If GUI is necessary, one has to work in an interactive session with X redirection (see *Users's guide*).

4.1. ABAQUS

ABAQUS (Abaqus Unified FEA) is a software package delivered by Dassault Systemes. It enables nonlinear analysis of systems with the use of finite-element method in complex engineering research. It is used in solid and fluids mechanics and to assess the durability of machines parts and construction, taking into account load, temperature, joints, possible collisions and other environmental conditions.

ABAQUS enables parallel calculations in MPI modes and threads.

4.1.1. ABAQUS/Standard

The Standard stands for general purposes modules for finite-element method analysis using implicit integration method. It is designed for solving of nonlinear processes. It is used to study static, dynamic, thermal and electrical events occurring at relatively low speed.

4.1.2. ABAQUS/Explicit

The Explicit is designed to solve dynamic tasks using explicit integration method of the nonlinear equation of motion. The program is used to analyze extreme mechanical, physical and thermal loads. It might be used in the endurance test of electronic devices, impacts, metal forming processes and cars crash tests.

4.1.3. ABAQUS/CAE

ABAQUS/CAE is the application used for pre-processing, editing, monitoring, diagnosing and visualization of the results of analysis conducted by ABAQUS/Standard or ABAQUS/Explicit. It combines the functionality of the preprocessor and post-processor.

4.1.4. Links

1. Description: <http://www.3ds.com/products-services/simulia/portfolio/abaqus/latest-release/>
2. Documentation: <http://dsk.ippt.pan.pl/docs/abaqus/>

References

- [1.] N. Aravas and E. C. Aifantis. On the Geometry of Slip and Spin in Finite Plastic Deformation. *International Journal of Plasticity*, 7(3):141–160, 1991.
- [2.] D. G. Ashwell and R. H. Gallagher, Editors. *Finite Elements for Thin Shells and Curved Members*. John Wiley and Sons, London, 1976.
- [3.] T. Ikeda. *Fundamentals of Piezoelectricity*. Oxford University Press, New York, 1990.
- [4.] M. Kleiber, H. Antúnez, T. D. Hien, and P. Kowalczyk. *Parameter Sensitivity in Nonlinear Mechanics*. John Wiley & Sons, 1997.
- [5.] O. C. Zienkiewicz. *The Finite Element Method*. McGraw-Hill, London, Third Edition, 1977.

4.2. ANSYS CFD

The full name of the package is Academic Research CFD. The package includes *Fluent* and *CFX* programs, and *Workbench 2* environment. *Gambit* and *TGrid* preprocessors might be treated as an addition to the package.

4.2.1. ANSYS Fluent

ANSYS Fluent software contains the broad physical modeling capabilities needed to model flow, turbulence, heat transfer, and reactions for industrial applications ranging from air flow over an aircraft wing to combustion in a furnace, from bubble columns to oil platforms, from blood flow to semiconductor manufacturing, and from

clean room design to wastewater treatment plants. Special models that give the software the ability to model in-cylinder combustion, aeroacoustics, turbomachinery, and multiphase systems have served to broaden its reach.

4.2.2. ANSYS CFX

ANSYS CFX software is a high-performance, general purpose fluid dynamics program that has been applied to solve wide-ranging fluid flow problems for over 20 years. At the heart of ANSYS CFX is its advanced solver technology, the key to achieving reliable and accurate solutions quickly and robustly. The modern, highly parallelized solver is the foundation for an abundant choice of physical models to capture virtually any phenomena related to fluid flow. The solver and its many physical models are wrapped in a modern, intuitive, and flexible GUI and user environment, with extensive capabilities for customization and automation using session files, scripting and a powerful expression language.

4.2.3. Workbench

ANSYS CFD solutions are fully integrated into the ANSYS Workbench platform. This environment delivers high productivity and easy-to-use workflows. Workbench integrates all your workflow needs (pre-processing, simulation and post-processing) as well as multiphysics functionality (fluid-structure interaction, electronic-fluid coupling). Workbench allows for automated and easy-to-set-up optimization or design exploration studies (for example, design of experiments, six sigma analysis).

4.2.4. Links

1. Description: <http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics>
2. Documentation: <http://www.ansys.com/Support/Documentation>

References

- [1.] P. Nitharasu and R. W. Lewis. Fundamentals of the Finite Element Method for Heat and Flow. Wiley & Sons, 2014.
- [2.] O. C. Zienkiewicz and R. L. Taylor. The Finite Element Method. Volume 3, Fluid Dynamics, 5th Edition, Butterworth – Heinemann, 2000.
- [3.] J. Donea and A. Huerta. Finite Element Methods for Flow Problems. Wiley & Sons, 2003.

- [4.] C. A. Felippa and K. C. Park. Staggered transient analysis procedures for coupled dynamic systems. *Comp. Meth. Appl. Mech. Eng.*, 26(1):61–111, 1980.
- [5.] <http://www.ansys.com/Resource+Library>

4.3. COMSOL

COMSOL Multiphysics is the software package to conduct analysis and simulations as well as solve user defined sets of non-linear partial differential equations using finite element method. It is widely used in different physics and engineering fields, particularly in the case of phenomena described by physics models. COMSOL offers interface with MATLAB, which fully integrates both programs and allows using all MATLAB software capabilities. COMSOL gives possibility of solving coupled systems of partial differential equations (PDE), as well as model and coupled phenomena analysis.

COMSOL cooperates with CAD type programs e.g. AutoCAD, Creo, SolidWorks. Livelink for SolidWorks module enables starting COMSOL simulation with GUI SolidWorks.

COMSOL capabilities can be extended by using modules in fields such as:

- acoustic,
- fluid dynamics,
- mechanics,
- electromechanics,
- chemical engineering,
- geophysics,
- navigation,
- applied mathematics.

COMSOL can be installed on different platforms such as Windows, Linux, MacOS. The program is capable of performing parallel calculations on multicore machines.

4.3.1. Links

1. <http://www.comsol.com>

References

- [1.] P. Nitharasu and R. W. Lewis. *Fundamentals of the Finite Element Method for Heat and Flow*. Wiley & Sons, 2014.
- [2.] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method*. Vol. 3, Fluid Dynamics, 5th Edition, Butterworth – Heinemann, 2000.

- [3.] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. Wiley & Sons, 2003.
- [4.] C. A. Felippa and K. C. Park. Staggered transient analysis procedures for coupled dynamic systems. *Comp. Meth. Appl. Mech. Eng.* 26(1):61–111, 1980.
- [5.] <http://www.comsol.com/papers-presentations>

4.4. DL_POLY CLASSIC

The DL_POLY_CLASSIC is a molecular dynamics simulation package developed at Daresbury Laboratory (UK) by W. Smith, T.R. Forester and I.T. Todorov. It is based on the package DL_POLY_2.

DL_POLY Classic can be executed as a serial or parallel application. The code achieves parallelisation using the Replicated Data strategy which is suitable for homogeneous, distributed-memory, parallel computers. The code is useful for simulations of up to 30,000 atoms with good parallel performance on up to 100 processors. In some circumstances, it can exceed or fail to reach these limits.

The copyright to the software is the property of the Science & Technology Facilities Council (STFC), and the code is made available under the BSD Licence, which permits open access to the source. No registration is required to obtain the software, and no support of any kind is offered or implied. Advice on using all DL_POLY packages may be obtained from the DL_POLY Online Forum. A full description of DL_POLY Classic may be obtained from the DL_POLY Classic User Manual (PDF).

4.4.1. Links

1. Description:
http://www.ccp5.ac.uk/DL_POLY_CLASSIC/
http://www.stfc.ac.uk/SCD/research/app/ccg/software/DL_POLY/44516.aspx
2. Forum:
http://www.stfc.ac.uk/CSE/randd/ccg/software/DL_POLY/25531.aspx
3. User manual:
http://www.ccp5.ac.uk/DL_POLY_CLASSIC/MANUALS/USRMAN.pdf
4. GUI manual:
http://www.ccp5.ac.uk/DL_POLY_CLASSIC/MANUALS/JavaGUI.pdf

References

- [1.] W. Smith and I. T. Todorov. A short description of DL_POLY. *Molecular Simulation*, 32(12–13):935–943, 2006.
- [2.] C. Kartsaklis, I. Todorov, and W. Smith, DL_POLY 3. Hybrid CUDA/OpenMP porting of the non-bonded force-field for two-body systems. *Abstracts of Papers of the American Chemical Society*, 2010.
- [3.] P. A. Cazade, P. Bordat, I. Baraille, R. Brown, W. Smith, and I. T. Todorov. DL_POLY_2 adaptations for solvation studies. *Molecular Simulation*, 37(1):43–52, 2011.
- [4.] <http://www.stfc.ac.uk/SCD/Publications/41153.aspx>

4.5. LAMMPS

LAMMPS is a classical molecular dynamics code, and an acronym for Large-scale Atomic/Molecular Massively Parallel Simulator. It has potentials for solid-state materials (metals, semiconductors) and soft matter (biomolecules, polymers) and coarse-grained or mesoscopic systems. It can be used to model atoms or, more generically, as a parallel particle simulator at the atomic, meso, or continuum scale.

The program runs on single processors or in parallel using message-passing techniques and a spatial-decomposition of the simulation domain. The code is designed to be easy to modify or extend with new functionality.

LAMMPS is distributed as an open source code under the terms of the GPL. It is distributed by Sandia National Laboratories, a US Department of Energy laboratory.

4.5.1. Links

1. Description: <http://lammps.sandia.gov/>

References

- [1.] C. Grindon, S. Harris, and T. Evans, et al. Large-scale molecular dynamics simulation of DNA: implementation and validation of the AMBER98 force field in LAMMPS. *Philosophical Transactions of The Royal Society A-Mathematical Physical and Engineering Sciences*, 362(1820):1373–1386, 2004.
- [2.] L. T. Kong, G. Bartels, C. Campana, C. Denniston, and M. Mueser. Implementation of Green's function molecular dynamics: An extension to LAMMPS. *Computer Physics Communications*, 180(6):1004–1010, 2009.
- [3.] D. B. Frantz, S. J. Plimpton, and M. S. Shephard, Software components for parallel multiscale simulation: an example with LAMMPS. *Engineering with Computers*, 26(2):205–211, 2010.
- [4.] <http://lammps.sandia.gov/papers.html>

4.6. LMG90

The LMG90 is a free open source software dedicated to multi-physics problems. The program uses NSCD (Non-smooth contact dynamics) method [1]. In general, the program covers hybrid discrete/finite element method. The very important features of the program are the possibility of analysis of compliant particles. The mechanical problem that is described with the DEM/FEM method can be coupled with other problems, namely, heat transfer and fluid flow. The program has got a new modular structure employing python, fortran90, c++, [2]. An example of a model containing irregular rigid particles is given in [3].

4.6.1. Links

1. https://subver.lmgc.univ-montp2.fr/trac_LMG90v2/

References

- [1.] M. Jean. The Non Smooth Contact Dynamics Method. *Computer Methods in Applied Mechanics and Engineering, special issue on computational modeling of contact and friction*, J.A.C. Martins, A. Klarbring (ed), 177:235–257, 1999.
- [2.] F. Dubois and M. Jean. The non-smooth contact dynamic method: recent LMG90 software developments and application. In *Lecture Notes in Applied Mechanics and Engineering: Analysis and Simulation of Contact Problems*, P. Wriggers, U. Nackenhorst (eds.), 27:375–378, Springer-Verlag, 2006.
- [3.] E. Postek, F. Dubois, and R. Mozul. Modeling of a collection of tensegrity particles with a non-smooth discrete element method. *38th Solid Mechanics Conference*. Warsaw, Poland, 27–31.08.2012. eds R. Pecherski, J. Rojek, P. Kowalczyk, pp. 28–29, 2012.

4.7. LS-DYNA

LS-Dyna is a general-purpose program, particularly useful when analyzing transient dynamics using finite element method in explicit formulation. The program enables the introduction of highly nonlinear models including the geometric and physical nonlinearities, contact problems and multiphysics coupling. There are more than one hundred material models available in the program e.g. linear and nonlinear elastic materials, viscous-elastic materials, elastic-plastic materials, damaged materials, composite materials.

The program can be used to simulation of wave phenomena, deformation of the structures during impacts, explosions, collisions, simulating technological processes such as those related to cutting and embossing. Additionally the program can perform the heat flow and electromagnetic analyses and compressible and incompressible fluid flow modeling coupled with construction deformation (FSI). Acoustic calculation module using boundary element method and SPH calculation module are also available.

LS-Dyna works on different platforms with different operating systems (Windows, Unix, MacOS).

4.7.1. Links

1. Description: <http://www.lstc.com/products/ls-dyna>

References

- [1.] T. Belytschko. Partitioned and Adaptive Algorithms for Explicit Time Integration. In *Nonlinear Finite Element Analysis in Structural Mechanics*, ed. by Wunderlich, W. E. Stein, and K. J. Bathe, , 572–584, 1980.
- [2.] K. J. Bathe. Finite Element Procedures in *Engineering Analysis*, Prentice-Hall, 1982.
- [3.] J. Rojek, E. Onate, and E. Postek. Application of explicit FE codes to simulation of sheet and bulk metal forming processes, *Journal of Material Processing Technology*, 80–81:620–627, 1998.
- [4.] <http://www.dynasupport.com/manuals/additional/ls-dyna-theory-manual-2005-beta>
- [5.] LS DYNA Conferences: <http://www.dynalook.com/>

4.8. MATHEMATICA

Wolfram Mathematica is the computing and visualizing environment being developed by Stephen Wolfram and Wolfram Research, Inc[1]. First version of the environment was publicized in 1988, the most recent version (May 2014) has got number 9.0.1. Apart from complex and automated numerical computing, the best known typical environment features are advanced possibilities of symbolic computation, which constitute the core of the package and programming language.

Full documentation of the recent environment version [2] and a brief overview of features [3] and application [4] are available on producer's websites. According to the producer environment key features are as follows:

- Automation,
- Integrated all-in-one platform,

- Hybrid symbolic-numeric methodology,
- Multiparadigm language based on LISP language,
- Built-in knowledge (e.g. meteorological, financial, mathematical, lexical sources available by Wolfram/Alpha).

Program interface enables interactive command entering, interaction with help system and displaying the results of calculations and their visualization. User's alternative interface is Wolfram Workbench, based on Eclipse environment- environment less interactive but more friendly towards programmers.

4.8.1. Links

1. Wolfram Research, Inc <http://www.wolfram.com>
2. Wolfram Mathematica Documentation Center
<http://reference.wolfram.com/mathematica/guide/Mathematica.html>
3. Mathematica Features <http://www.wolfram.com/mathematica/features/>
4. Wolfram Mathematica Overview
<http://www.wolfram.com/mathematica>
5. Wolfram Workbench <http://www.wolfram.com/products/workbench/>
6. Mathematica Resources <http://www.wolfram.com/mathematica/resources/>
7. Get Started with Mathematica
<http://www.wolfram.com/support/learn/get-started-with-mathematica/>

References

- [1.] R. Cooper and J. Casanova. 2-Dimensional atomic and molecular-orbital displays using Mathematica. *Journal of Chemical Education*, 66(6):487–488, 1991.
- [2.] N. I. Ioakmidis. Application of Mathematica to the direct seminumerical solution of finite element problems, *Computers & Structures*, 45(5–6):833-39, 1992.
- [3.] M. D. Mikhailov. Finite difference method by using Mathematica. *International Journal of Heat and Mass Transfer*, 37:375–379, 1994.
- [4.] <http://library.wolfram.com/infocenter/search/>

4.9. MATLAB

MATLAB is a numerical computing environment which uses its high level programming language. The name MATLAB comes from the phrase “Matrix Laboratory”, because the original purpose of the program was matrix operations. MATLAB also enables drawing two dimension and three dimension function graphs and visualization of results in the form of static images or animations.

The origins of the program go back to seventies. Program inventor Cleve Moler from the University of New Mexico wanted to give his students access to Fortran libraries to perform matrix operations without them having to learn programming in Fortran. In 1983, the program was rewritten in C and in 1984 MATLAB developing company called MathWorks was established.

Work in MATLAB can be done in interactive mode through console window or through calling scripts and functions. MATLAB treats all entered or declared data as a matrix – a single number is a matrix of size 1x1. Variables can be assigned without declaring their type. MATLAB recognizes variables automatically, only the correct calling is needed.

MATLAB software language is a high level one. The language syntax is based on C language syntax. It includes known from other languages control commands (if, for, while, switch) and allows object-oriented programming.

4.9.1. Toolbox

Toolboxes are set of functions (m-files form) to solve advanced problems in specific domains. Those toolboxes enlarge capabilities of MATLAB and are also written by other software producers.

Toolbox examples:

- Analysis and financial calculations,
- Fuzzy Logic Toolbox – designing and diagnostics intelligent control systems based on fuzzy logic,
- Image Processing Toolbox – Image processing,
- Parallel Computing Toolbox – Parallel programming and calculation on GPU,
- Partial Differential Equation Toolbox – numerical solving partial differential equation using finite element method,
- Simulink – modeling, simulation and analysis of dynamic systems, includes user's graphic interface allowing models construction in the form of diagrams.

4.9.2. Parallel programming

Parallel Computing Toolbox (PCT) allows to accelerate calculations by using parallel processing. It can be achieved in several ways:

- parallel processing algorithm in built-in functions,
- programming language structure – parfor loop, distributed arrays, SPMD,
- parallel work on local CPU cores,

- MATLAB compute cluster – MATLAB Distributed Computing Server,
- calculations on GPU.

4.9.3. Links

1. Main page: <http://www.mathworks.com/>
2. Documentation: <http://www.mathworks.com/help/>

References

- [1.] J. H. Wilkinson and C. Reinsch. *Handbook for Automatic Computation*. Springer, 1971.
- [2.] <http://www.mathworks.com/company/newsletters/articles/the-origins-of-matlab.html>
- [3.] http://www.tech.plym.ac.uk/spmc/links/matlab/matlab_papers.html

4.10. MUMPS

The MULTifrontal Massively Parallel sparse direct Solver (MUMPS) is a library for the solution of linear sparse unsymmetric systems of equations. The solver is designed for parallel distributed computers. The solver is well suited for the finite element techniques. The matrices can be supplied in assembled and elemental formats. It is possible to solve the systems of equations with several right hand sides. The library is written in C and Fortran 90 languages. It has interfaces to C, Fortran 90 and Matlab. The solver can use both the RAM and disk memories. It has several orderings: *AMD*, *AMF*, *PORD*, *METIS*, *PARMETIS*, *SCOTCH*, *PT-SCOTCH*.

The solver has been presented during the post-graduate course given in IPPT, “Introduction to high-performance computing (KDM-HPC) via applications”.

4.10.1. Links

1. <http://mumps.enseeiht.fr/>
2. http://mumps.enseeiht.fr/ud_2013.php
3. <http://mumps.enseeiht.fr/index.php?page=apps>

References

- [1.] P. R. Amestoy, I. S. Duff, J. Koster, and J. Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal of Matrix Analysis and Applications*, 23(1):15–41, 2001.

- [2.] P. R. Amestoy, A. Guermouche, J. Y. L'Excellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2): 136–156, 2006.
- [3.] E. Postek, R. W. Lewis, and D. T. Gethin. Finite element modelling of the squeeze casting process. *International Journal of Numerical Methods for Heat and Fluid Flow*, 18(3–4):325–355, 2008.
- [4.] E. Postek. Concept of an Agent-stress Model of a Tissue. *Technische Mechanik*, 32:518–529, 2012.

4.11. OPENFOAM

The OpenFOAM® (Open Field Operation and Manipulation) CFD Toolbox is a free, open source CFD software package which has a large user base across most areas of engineering and science, from both commercial and academic organisations. OpenFOAM has an extensive range of features to solve anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to solid dynamics and electromagnetics. It includes tools for meshing, notably snappyHexMesh, a parallelised mesher for complex CAD geometries, and for pre- and post-processing. Almost everything (including meshing, and pre- and post-processing) runs in parallel as standard, enabling users to take full advantage of computer hardware at their disposal.

The program is developed and maintained by ESI-OpenCFD Group.

4.11.1. Links

1. <http://www.openfoam.com/>
2. <http://www.esi-group.com/company/events/2014/2nd-openfoam-user-conference-2014>
3. http://openfoamwiki.net/index.php/Main_Page
4. <http://foamcfd.org/>
5. <http://www.cfd-online.com/Forums/openfoam/>
6. http://homepages.cae.wisc.edu/~hessel/openfoam/Resources/index_OpenFOAMResources.html

References

- [1.] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object orientated techniques. *Computers in Physics*, 12(6):620–631, 1998.
- [2.] [http://openfoamwiki.net/index.php/Main_ContribPapers#The original paper outlining OpenFOAM](http://openfoamwiki.net/index.php/Main_ContribPapers#The_original_paper_outlining_OpenFOAM)

4.12. PARAVIEW

ParaView is an open-source, multi-platform data analysis, and visualization application. ParaView users can quickly build visualizations to analyze their data using qualitative and quantitative techniques. The data exploration can be done interactively in 3D or programmatically using ParaView's batch processing capabilities.

ParaView was developed to analyze enormous datasets using distributed memory computing resources. It can be run on supercomputers to analyze datasets of exascale size as well as on laptops for smaller data.

The main project sponsor is Kitware company. Since the beginning of the project, Kitware has successfully collaborated with Sandia, Los Alamos National Laboratories, the Army Research Laboratory and various other academic and government institutions to continue development.

4.12.1. Links

1. <http://www.paraview.org/>
2. <http://www.kitware.com/>

References

- [1.] A. Henderson, J. Ahrens, and Ch. Law. *The ParaView Guide*. Kitware Inc., 2004, Clifton Park, NY.
- [2.] <http://www.paraview.org/publications/>

4.13. TECPLOT 360

Tecplot 360 is the software package used to visualize the results of numerical simulations in the field of CFD. It cooperates with Fluent program, it can also read data from Excel, FLOW3D and ASCII files.

4.13.1. Links

1. <http://www.tecplot.com/products/tecplot-360/>
2. <http://www.cfd-online.com/Forums/tecplot/>

References

- [1.] R. Hynds. TECPLOT. *New Scientist*, 142:44–44, 1994.
- [2.] K.R. Foster. Visualise this! With Tecplot 10, what you've got is what you see. *IEEE Spectrum*, 41(8):52–54, 2004.
- [3.] <http://www.tecplot.com/white-papers/>

4.14. VMD

VMD (Virtual Molecular Dynamics) is a program for visualization, modeling and analysis of biological systems such as proteins, nucleic acids, lipid bilayer assemblies, etc. The program can visualize more general molecules as it can read directly standard Protein Data Bank (PDB) files. VMD provides a wide variety of methods for rendering and coloring a molecule: simple points and lines, CPK spheres and cylinders, licorice bonds, backbone tubes and ribbons, cartoon drawings, and others. It can be used to animate and analyze the trajectory of a molecular dynamics (MD) simulation. In particular, VMD can act as a graphical front end for an external MD program by displaying and animating a molecule undergoing simulation on a remote computer. It can be integrated with NAMD (parallel molecular dynamics code designed at UIUC) and DL_POLY. VMD program supports all major computer platforms as well as GPU accelerated computation.

The program is developed and maintained in the Theoretical and Computational Biophysics Group at the University of Illinois at Urbana-Champaign (UIUC). The papers published by the members of the group were referenced 70850 times in the years 1972-2014. The highly cited papers are for example [1] (citations: 12743) , [2] (1823), [3] (citations: 1035).

4.14.1. Links

1. <http://www.ks.uiuc.edu/Research/vmd/>
2. <http://www.sas.upenn.edu/~robertjo/html-physics/vmd/>
3. <http://www.physics.upenn.edu/~robertjo/vmd/>
4. <http://www.buffalo.edu/ccr/support/software-resources/Visualization/vmd.html>
5. <http://www.ks.uiuc.edu/Research/namd/>

References

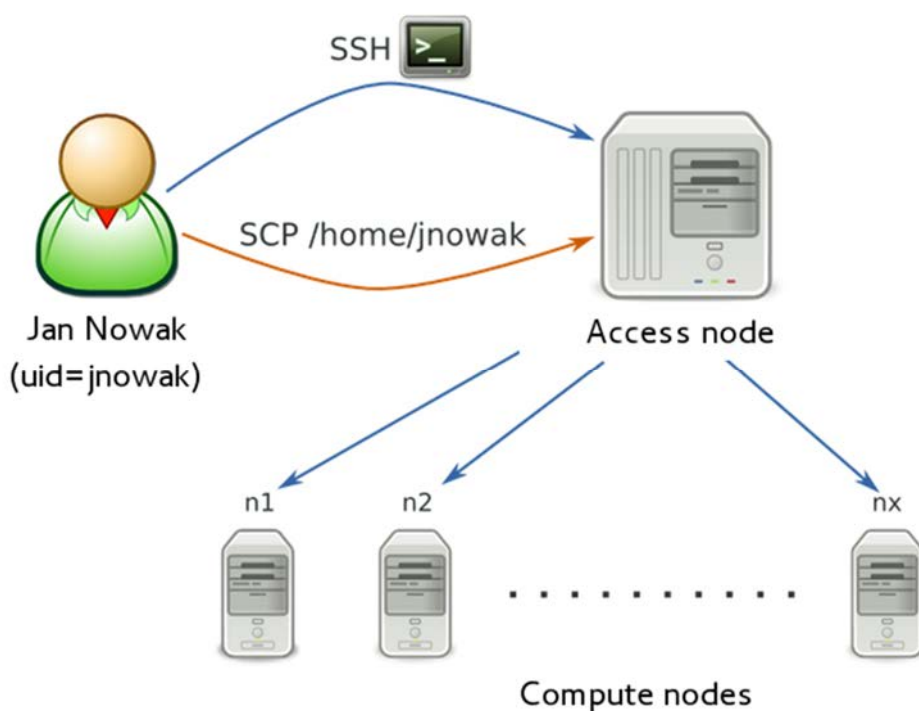
- [1.] W. Humphrey, A. Dalke, and K. Schulten. VMD-Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996.
- [2.] L. Kalé, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, and K. Schulten. NAMD2: Greater scalability for parallel molecular dynamics. *Journal of Computational Physics*, 151(1):283–312, 1999.
- [3.] M. Sarikaya, C. Tamerler, A.K.-Y. Jen, K. Schulten, and F. Baneyx. Molecular biomimetics: nanotechnology through biology. *Nature Materials*, 2:577–585, 2003.
- [4.] <http://www.ks.uiuc.edu/Publications/Papers/>
- [5.] <http://www.sas.upenn.edu/~robertjo/html-physics/publications/>

Grafen cluster user's guide

Rafał Gawlik, Eligiusz Postek

5.1. Introduction

From the user's point of view Grafen cluster consists of HPC compute cluster and other servers which work on virtual machines and are logically independent. HPC cluster is visible for user as follows:



Computers in the cluster are called *nodes*. They are available for the user through access node. Access node is a “bridge” between the user and other nodes (other sources) of the cluster. Login and files transfer is possible only through the access node.

5.2. How to use the system

All researchers of IPPT PAN are invited to use cluster system. Separate users' accounts systems are used for compute nodes and virtual machines. Accounts are created by employee of Computer Network Service (DSK).

Using compute resources one must be aware of some additional rules:

- To start a program one must log in access node, next use task manager, which allows assignment of needed resources (CPU, RAM) on compute nodes,
- Tasks requiring contact with the user (e.g., compilations) should be started in interactive manager mode (*qsub -I*),
- Files should be placed in proper location – source codes in home directory, temporary data in *\$TMPDIR*.

5.3. Virtual servers

Application cluster provides virtual machines under the control of VMware system. Special attention should be paid to Windows 2008 server and MATLAB compute server.

5.3.1. Windows 2008 server

The server allows remote use of graphic and scientific software in Microsoft Windows environment. User account is independent from other systems and it is created by the administrator. Access to the server is possible through connection to *remote desktop* from Windows or *rdesktop* from Linux.

Available software:

- MATLAB 2011b,
- Origin 9.0,
- Inkscape,
- GIMP,
- CorelDraw 12,
- Acrobat Professional,
- Microsoft Office 2010 Professional,
- other programs on demand.

5.3.2. MATLAB Computing Server

MATLAB Distributed Computing Server allows to launch tasks requiring parallel or distributed processing on dedicated computing cluster. The cluster currently consists of sixteen compute cores on which MATLAB computational modules (workers) are running.

Working with MATLAB computing cluster is to prepare the task locally using MATLAB program with installed Parallel Computing Toolbox and then send it to cluster in order to perform calculations. Tasks can be performed in interactive or batch mode.

Thanks to its own queuing system called MATLAB Job Scheduler, the cluster can be used by many users at the same time. The cluster is installed on the virtual machine controlled by Linux and the access to it is also available from Linux computer. Detailed instruction of use is available on DSK websites.

5.4. HPC system access

To work with HPC cluster three components are required:

- SSH client – allows to log in on access node,
- SCP client – allows copying files on access node,
- X Server – allows forwarding X11 protocol needed to work in graphic mode.

Access node is located on *headnode.kevlar.ippt.pan.pl* (IP 10.100.255.73). Login to all succeeding system nodes is done *without the password*.

After logging in tasks manager has to be used to get the access to resources of the cluster.

NOTICE – Do not log in or manually start any processes on compute nodes which have not been allocated by the task manager. Manual tasks running causes interference to other users processes.

5.4.1. LOG IN – SSH

The SSH client is installed by default in Linux and MacOS system. To connect with the cluster a command similar to the following should be entered in a system console:

```
$ ssh -X username@headnode.kevlar.ippt.pan.pl
```

Then the user is asked for a password to access the cluster. *X* parameter means automatic forwarding and is needed when the user intends to use the application in the graphic mode.

Windows – there is no default SSH client, it has to be downloaded and installed. *PuTTY* is a recommended SSH client. If the user plans to use X11 forwarding, installation of X server is required, *Xming* server is recommended there.

5.5. Files transfer – scp

SCP is the easiest way to transfer files to and from the access node.

5.5.1. Linux and Mac OS

SCP protocol is installed by default. The command must be entered in system console.

Copying the local file on access node to home directory:

```
$ scp local_file user@headnode.kevlar:
```

Copying the file from a node to local disc:

```
$ scp user@headnode.kevlar:/path/file_on_server
```

In both cases the user password is required. Copying using file manager is also possible in selected graphical environment (e.g., Gnome, KDE, XFCE).

5.5.2. Windows

To copy the files *WinSCP* program is recommended

5.6. Files transfer – sftp

Linux – two modes are possible: *intactive mode* and *batch mode*.

5.6.1. Interactive mode

Interactive mode is started by command *sftp user@remotehost*:


```
$ sftp user@headnode.kevlar
user@headnode.kevlar's password: (type password here)
Connected to headnode.kevlar.
sftp>
```

After *sftp* session opening commands known from *ftp* program: *put*, *get*, *pwd*, *cd*, *lcd* are available. To display the list of available commands enter *help* in *sftp* session.

5.6.2. Batch mode

Method 1: Downloading file or files from the server to local computer. Wildcard characters (*, ?).can be used.

```
$ sftp user@remotehost:/remote/filename localfilename
```

Method 2: Creating commands file of *sftp* program, that will be executed one after the other.

```
$ sftp -b batchfile user@remotehost
```

More information can be found in documentation – *man sftp*.

5.7. Task manager

5.7.1. Available resources

To check the size of available memory and computing cores the user can perform the task manager command *pbsnodes*. The most popular callings of this command are as follows:

- *pbsnodes -a* shows available resources on all nodes,
- *pbsnodes <node name>* shows available resources of selected node.

The example of above mentioned command:

```

$ pbsnodes n4
n4
    state = free
    np = 12
    properties = kevlar,intel,mpi
    ntype = cluster
    status = rectime=1327145281,va-
rattr=, jobs=, state=free, net-
load=804429755, gres=, loadave=0.00, ncpus=12,
    physmem=24685132kb, availmem=32322812kb, tot-
mem=32685460kb, idletime=563910, nusers=0, nsessions=? 15201,
    sessions=? 15201, uname=Linux n4 2.6.18-238.19.1.el5
#1 SMP Fri Jul 15 00:48:58 EDT 2011 x86_64, ophys=linux
gpus = 0

```

That means node n4 :

- is switched on and ready to work (*state=free*),
- has 12 computing cores (*ncpus=12*),
- is not loaded (*loadave=0.00*),
- has 24 GB RAM memory (*physmem=24685132kb*),
- has no GPU accelerator (*gpus=0*).

NOTICE – The user may encounter offline nodes. It means that the node is temporarily excluded from the computing system, e.g., due to service.

5.7.2. Starting the task

To start the tasks on nodes, the command `qsub` is used. Detailed documentation can be found in the user's guide displayed by using `qsub` command or on *TORQUE* documentation website. The basic parameter (given at the end) is a script to perform, the most important options are as follows:

- `-N <task_name>`

allows to give name the task which will be used to monitor the task;

- `-l nodes=<number_of_nodes>:ppn=<numbers_of_cores>`

reserves for the task defined number of computing nodes and a defined number of cores in each;

- `-l walltime=<hours>:<minutes>:<seconds>`

reserves specified computing time, after which the task is removed;

- `-I`

performs interactive task – at the moment of nodes allocation the user is redirected

to the console of the first one and there may perform the task manually; in that mode tasks as compilation and data pre- and postprocessing should be performed;

- *-X (optionally -vDISPLAY)*

Starts X graphical system to interactive work

- *-l pmem=<memory_reserved>*

reserves specified memory quantity per core in megabytes.

Examples:

<i>qsub -N new_task script.sh</i>	Add task of preparing the script skrypt.sh to the calculating queue. The name of the task is new_task.
<i>qsub -l nodes=10:ppn=12 -l walltime=0:10:00 task.sh</i>	The same as above, reserves 10 nodes 12 cores each for computing, the task will last max. 10 minutes.
<i>qsub -I -X nodes=4</i>	Performs interactive task – the user is given a console on the first node allocated to the task. Additionally X redirection is running.
<i>qsub -l nodes=12:ppn=12 -l pmem=1GB</i>	Reserves 12 nodes, 12 cores each, each core 1 GB RAM memory
<i>qsub -N new_task script.sh</i>	Add task of preparing the script skrypt.sh to the calculating queue. The name of the task is new_task.

Task parameters can be also given in the content of the current script. Comments starting from letters PBS are automatically recognized by the queuing system. Argument names are left unchanged. The script below might be considered as an example:

```
#!/bin/sh
#PBS -N myTask
#PBS -l nodes=1:ppn=12,walltime=00:01:00

# Prepare data
cd $TMPDIR

echo Running...

cd ~/mp_linpac/bin_intel/intel64
mpirun -np $PBS_NP -f $PBS_NODEFILE ./xhpl_intel64

# Cleaning
cp $TMPDIR/input_file ~/output_file
```

That kind of scripts can be started by `qsub script.sh` command and all needed parameters can be found there. The meaning of `$TMPDIR` directory catalogue has been explained in *Mass Storage* part while the method of launching MPI task has been explained in *MPI Task* part.

5.7.3. Accurate resources identification

The above mentioned `-l nodes=<number_of_nodes>;ppn=<number_of_cores>` type request syntax allows to define the number of computing nodes and used cores in each of them. However, sometimes it is not enough, e.g., sending the task requires 10 separate nodes with 1 core each and one node with six cores. In this situation, using the syntax showed above, request for 11 nodes with 6 cores each should be made, which would lead to resources waste, or the task would be divided into two separate tasks, which is not always possible and trouble-free. The situations, where not all computing nodes are identical, are also possible (e.g., different type of processor), and that is why we would like to have one preferable node being among nodes assigned to our task. Because of that in task manager there is an option to use so-called *Multi-Req* function which means that flexible resources request can be formulated.

In *Multi-Req* resources, request takes the form of a list of individual requests separated by pluses.

```
qsub -l nodes=<request1>+<request2>+...+<requestN>
```

where `<request>` takes one of two forms:

- `<number_of_nodes>;ppn=<number_of_cores>` or
- `<node_name>;ppn=<number_of_cores>`,

to point the preferred node.

For example, the following requests are valid:

- `qsub -l nodes=10:ppn=1 task.sh`
- `qsub -l nodes=10:ppn=1+1:ppn=6 task.sh`
- `qsub -l nodes=10:ppn=1+n13:ppn=6 task.sh`

First of them uses current convention to request 10 nodes one processor each. The second command add request for additional node with six cores. The third command accomplish the next possibility – choice of the preferred computing node (here – `n13`) to the part of the task that needs 6 cores.

5.7.4. Parallel programs

Task file forwarded to execute, apart from parameters of the task, should also include commands to execute. Note that those commands would be executed only on first of allocated nodes. In the case when our task consists of many computing nodes, the valid command should be entered.

Interactive mode

If we have chosen the interactive mode, we can just log in on appropriate nodes through ssh and write our commands there. The nodes allocated to our task can be recognized by displaying `$PBS_NODES` file (the names of nodes are multiplied according to the number of allocated tasks).

```
[username@headnode username]$ qsub -I -l nodes=3:ppn=2
qsub: waiting for job 5559.master1.kevlar to start
qsub: job 5559.master1.kevlar ready

[username@n8 ~]$ cat $PBS_NODEFILE
n8
n8
n7
n7
n6
n6
[username@n8 ~]$ ./task.bin 1 &
[1] 10794
[username@n8 ~]$ ssh n7
[username@n7 ~]$ ./task.bin 2 &
[1] 5164
[username@n7 ~]$ exit
[username@n8 ~]$ ssh n6
[username@n6 ~]$ ./task.bin 3 &
[1] 4489
[username@n6 ~]$ exit
[username@n8 ~]$
```

Automatically - pdsh

The command *pdsh* should be used to avoid manual forwarding of tasks in the above mentioned way. It allows remote execution of the specified commands on chosen nodes. For example, to display the time of work for *n2*, *n3*, *n4* and *n7* nodes without logging on them, the following command should be entered:

```
[username@headnode username]$ pdsh -w n[2-4],n7 uptime
n7:  17:22:42 up 66 days,  6:00,  0 users, load average:
7.99, 7.97, 7.99
n4:  17:22:50 up 66 days,  6:00,  0 users, load average:
0.06, 0.07, 0.00
n3:  17:21:56 up 66 days,  6:00,  0 users, load average:
0.00, 0.00, 0.00
n2:  17:22:36 up 66 days,  6:00,  0 users, load average:
0.00, 0.00, 0.00
```

For greater convenience, the *pdsh* command is integrated with queuing manager. It means that when issuing the command from the computational task, we do not have to enter the names of nodes – chosen program would be executed on nodes allocated to the task.

Examples:

```
[username@headnode ~]$ qsub -I -l nodes=3:ppn=2
qsub: waiting for job 5560.master1.kevlar to start
qsub: job 5560.master1.kevlar ready

[username@n24 ~]$ pdsh uptime
n9:  20:16:50 up 66 days,  8:54,  0 users, load average:
7.08, 7.02, 7.01
n8:  20:15:42 up 66 days,  8:53,  0 users, load average:
0.00, 0.00, 0.00
n24: 20:16:26 up 42 days,  6:15,  0 users, load average:
0.04, 0.01, 0.00
[username@n24 ~]$
```

This mechanism works in the same way in noninteractive tasks, from the level of task script.

MPI tasks

The list of available MPI libraries can be displayed with the *module avail* command, to load MPI library execute *load module <mpi-module-name>* command (see section 5.9 *Environment Modules* for all module commands). To check which *MPI* library is active, *which mpirun* command can be executed.

When we use MPI library, the number of processors that we would like to run and computers on which they are to be launched should be specified.

```
$ mpirun -np <number_of_processes> -f <hostnames_file> <program_name>
```

Here also, the integration with *Torque* can be used. It provides system variables, set on proper values based on task parameters. Thanks to that all MPI programs calling may look the same:

```
$ mpirun -np $PBS_NP -f $PBS_NODEFILE <program_name>
```

5.8. Task monitoring

5.8.1. Task state

To check what is the state of our task *qstat* command can be used. It is the task manager's task. The most popular calling of this task is as follows:

- *qstat* shows the state of all user's tasks in abbreviated form,
- *qstat -f <job_id>* shows detailed information about chosen task.

The example result of the first command is showed below:

```
[tester@headnode ~]$ qstat
Job id          Name          User          Time Use S Queue
-----
293.master1    listing      tester        00:00:00 C batch
294.master1    task-wait1   tester        00:02:13 R batch
295.master1    task-wait2   tester         0 R batch
296.master1    task4        tester         0 R batch
297.master1    task5        tester         0 Q batch
```

The meaning of the columns is as follows:

- *Job id* – unique task ID, created on the basis of ordinary number and host name from which it was called,
- *Name* – name given by the user,
- *User* – name of the user who commissioned the task,
- *Time Use* – quantity of time used by the processor,
- *S* – State of the task (*Q* – waiting in the queue, *R* – running, *C* – completed),
- *Queue* – name of the queue where the task.

We see then that the above print informs us about following tasks:

- *listing* – the time of the processor is not occupied and the task is running),
- *task-wait* – took 2 minutes and is completed,
- *task-wait2* – running,
- *task4* – running,
- *task5* – waiting for resources to be free to use.

Detailed information about chosen task contains much more parameters.

```
[tester@headnode ~]$ qstat -f 294.master1
Job Id: 294.master1
  Job_Name = j-walker-299_310
  Job_Owner = tester@headnode.kevlar
resources_used.cput = 5790:14:54
resources_used.mem = 1377128kb
resources_used.vmem = 1414100kb
resources_used.walltime = 483:03:04
job_state = R
queue = long
server = master1.kevlar
Checkpoint = u
ctime = Tue Jul 15 11:31:00 2014
Error_Path = headnode.kevlar:/mnt/lustre/home/mkochan/s
             storage/walker/run/j-walker-299_310.e734424
exec_host = n13/0+n13/1+n13/2+n13/3+n13/4+n13/5+n13/6+n
            13/7+n13/8+n13/9+n13/10+n13/11
exec_port = 15003+15003+15003+15003+15003+15003+15003+1
            5003+15003+15003+15003+15003
```



```

Hold_Types = n
Join_Path = n
Keep_Files = n
Mail_Points = a
mtime = Tue Jul 15 11:31:11 2014
Output_Path = headnode.kevlar:/mnt/lustre/home/tester/s
               storage/walker/run/j-walker-299_310.o734424
Priority = 0
qtime = Tue Jul 15 11:31:00 2014
Rerunable = True
Resource_List.nodect = 1
Resource_List.nodes = 1:ppn=12
Resource_List.walltime = 20000:00:00
session_id = 2714
etime = Tue Jul 15 11:31:00 2014
submit_args = -q long -N j-walker-299_310 -l walltime=2
              0000:00:00 -l nodes=1:ppn=12
start_time = Tue Jul 15 11:31:11 2014
Walltime.Remaining = 70260970
start_count = 1
fault_tolerant = False
job_radix = 0
submit_host = headnode.kevlar

```

The most important (except the ones described above) are:

- *resources_used.cput* – used time of processor running,
- *resources_used.mem* – used memory,
- *resources_used.walltime* – total time of the task,
- *Error_Path, Output_Path* – path to the files with task results (see section 5.8.3 *Output data*),
- *exec_host* – names of the nodes on which the task was running (here, two cores on each node n1-n13),
- *submit_args* – parameters for which the task was commissioned,
- *start_time* – the beginning time of the task.

5.8.2. Deleting the task

We can delete any task at any time regardless of its current state with *qdel <job_id>* command. In the case of waiting task, it is deleted from the queue, running task will be stopped. In both cases task achieves *C* state – completed.

5.8.3. Output data

The task can pass the results of its actions in different ways. It is easy when it is done by files – users' home directories on all nodes are shared, so after the completion of program's action, one check the results in working directory as it is done on standard PCs. However, there is a need to additionally describe data transferring on standard output and error output Normally the data is displayed on the console screen, however in case of task manager the program can be transferred to the execution while the user is away, e.g., at night.

Because of that to each task additional files, containing output data in working directory are created; task names are as follows:

- *<task_name>.o<task_number>* – standard program output,
- *<task_name>.e<task_number>* – standard error output.

In the previous example for the task named *task-wait1* and using *294.master1* ID, *task-wait1.o294* and *task-wait1.e294* files have been created. Obviously it does not include tasks run in interactive mode (*qsub -I*), where standard outputs are displayed directly on the screen.

5.9. Environment modules

Environment Modules collect environment settings for the program. Environment variables can be set, and necessary modules loaded prior to loading the program. Modules can be loaded, unloaded or switched on-the-fly.

The following section presents module management commands with sample output.

List of available modules *module avail*

```

$ module avail

----- /usr/share/Modules/modulefiles -----
dot          module-info null
module-cvs   modules      use.own

----- /usr/share/Modules/modulefiles_extra/mpi -----
mpich2/mpich2-1.2.1      openmpi/1.6.5-gcc(default)
mvapich2/mvapich2-1.8.1  openmpi/1.6.5-gcc_4.8.2
openmpi/1.4.3-gcc        openmpi/1.6.5-intel

----- /usr/share/Modules/modulefiles_extra/compilers -----
PE-gcc          gcc-4.8.2
PE-intel        intel/14.0(default)

----- /usr/share/Modules/modulefiles_extra/lib -----
scalapack/2.0.2-gcc  scalapack/2.0.2-intel

----- /usr/share/Modules/modulefiles_extra/tools -----
dl_poly/classic_1.9    python/2.7.3(default)
openfoam/2.3.0         python/3.3.0
paraview/4.1.0         vmd/1.9.1

to see module info - module whatis module-name

```

Module information *module whatis <module-name>*

```

$ module whatis python
python          : Sets up Python 2.7.3 as default in yo
ur environment

```


5.10. Mass storage

All computing nodes are connected to shared files system called *LUSTRE*. Nodes have not got local disks, and they start from *LUSTRE* resources, so they have got a homogenous configuration. If the administrator does not decide otherwise; all computing nodes start from the same operating system image.

- */home*
Users home directories. Source files, executable parallel file, output data and input data should be located there. Home directories are synchronized and displayed in the homogenous way on all cluster's nodes.
- */scratch*
Temporary fast computational space on SSD discs. In that part, there is no division on users and to avoid conflicts the data should be put in an individual directory created for each task, its name in `$TMPDIR` system variable. After task completion, the data is no more available.
- */tmp*
Ramdisk directory (virtual files system located on RAM memory of a node).

NOTICE! Do not use /tmp without administrator's direct instruction.

Starting applications on cluster – examples

Rafał Gawlik

6.1. Introduction

Easy way to start tasks on the cluster is preparation of the script that includes transferred to queuing system information about needed resources (line start from *#PBS*) and commands starting the computing application or compiled program.

Cluster is not designed to work in an interactive mode so the script should be prepared earlier on user's computer. Programs and scripts prepared to work on the cluster should not use functions which require displaying graphical windows, because programs launched in the queue work only in text mode.

6.2. ANSYS Fluent Example

Examples can be taken from: */home/_EXAMPLE/fluent_job* directory.

Files

fluent_job directory contains following files:

- *airfoil.cas*,
- *airfoil.dat*,
- *journal.jou*,
- *fluent_job.sh* script starting the simulation.

Script

```
#!/bin/bash -l
#PBS -N fluent_job
#PBS -l nodes=1:ppn=4
#PBS -l walltime=00:05:00

USER_NAME=user #change to your user name
cd /home/$USER_NAME/fluent_job

journal=journal.jou
FLUENTSOLVER=2ddp

# DON'T CHANGE BELOW THIS LINE
rm -f pnodes
cat $PBS_NODEFILE | sort > pnodes
export ncpus=`cat pnodes | wc -l`

fluent150 -g -pmmpi-auto-selected -mpi=hp $FLUENTSOLVER -t$ncpus -cnf=pnodes -ssh -i $journal > fluent.log
```

- *#PBS -N fluent_job*
#PBS -l nodes=1:ppn=4
#PBS -l walltime=00:05:00
Forwarding parameters to the queuing system: task name - *-fluent_job*, number of nodes – 1, number of cores – 4, prescribed time: 5 minutes.
- *cd /home/\$USER_NAME/fluent_job*
Change working directory.
- *journal=journal.jou*
FLUENTSOLVER=2ddp
rm -f pnodes
cat \$PBS_NODEFILE | sort > pnodes
export ncpus=`cat pnodes | wc -l`
Setting program starting parameters.
- *fluent150 -g -pmmpi-auto-selected -mpi=hp \$FLUENTSOLVER -t\$ncpus -cnf=pnodes -ssh -i \$journal > fluent.log*
Launching the program with redirection of standard output to *fluent.log* file.

Adding the task to the queue

Prepared scripts is added to the queue by a following command:

```
$ qsub fluent_job.sh
```

As the result of program action the output data file will be created:

- *airfoil-final.dat*.

and files:

- *fluent_job.eXXXXXX* standard error file,
- *fluent_job.oXXXXXX* standard output file (empty),
- *fluent.log* redirected program output,
- *pnodes*.

6.3. COMSOL Example

Example can be downloaded from: */home/_EXAMPLE/comsol_job* directory.

Files

comsol_job direcorey contains:

- *mapk-for-kevlar.mph* data file for simulation,
- *libegf-b.so* a library used in simulation,
- *run.sh* a script starting a simulation.

Script

The script starts the calculation in COMSOL program.

```
#!/bin/bash
#PBS -N comsol_job
#PBS -l nodes=1:ppn=12

USER_NAME=user #change to your user name

cd /home/$USER_NAME/comsol_job

/home/_PROGS/bin/comsol43b -ckl batch -inputfile
/home/$USER_NAME/comsol_job/mapk-for-kevlar.mph -outputfile
```

```
/home/$USER_NAME/comsol_job/mapk-for-kevlar.out.mph
-batchlog /home/$USER_NAME/comsol_job/mapk-for-kevlar.out.log
```

- *#PBS -N comsol_job*
#PBS -l nodes=1:ppn=12
 Forwarding parameters to the queuing system: task name – *comsol_job*, number of nodes – *1*, number of cores – *12*.
- *cd /home/\$USER_NAME/comsol_job*
 Change working directory.
- */home/_PROGS/bin/comsol43b -ckl batch*
-inputfile /home/\$USER_NAME/comsol_job/mapk-for-kevlar.mph
-outputfile /home/\$USER_NAME/comsol_job/mapk-for-kevlar.out.mph
-batchlog /home/\$USER_NAME/comsol_job/mapk-for-kevlar.out.log
 Starting the COMSOL program with parameters:
 - *-inputfile* *.../mapk-for-kevlar.mph* input data,
 - *-outputfile* *.../mapk-for-kevlar.out.mph* output data,
 - *-batchlog* *.../mapk-for-kevlar.out.log* log.

Adding the task to the queue

Prepared scripts are added to the queue by a following command:

```
$ qsub comsol_job.sh
```

Apart from output data and log, as the result of program action the following files will be created:

- *comsol_job.eXXXXXX* standard error file,
- *comsol_job.oXXXXXX* standard program output,
- *mapk-for-kevlar.out.mph.status* computational task status.

6.4. LS-DYNA Example

Example can be downloaded from: */home/_EXAMPLE/lsdyna_job*.

Files

lsdyna_job directory contains files generated by *LS-PrePost* program, which acts as pre- and postprocessor:

- *billet.forge.209.d*,
- *billet.forge.209.i*,
- *billet.forge.209.k*.

and the script:

- *lsdyna_job.sh*.

Script.

The script starts calculation in LS-DYNA program in parallel mode using *mpi* *MVAPICH* package.

```
#!/bin/sh
#PBS -N lsdyna_job
#PBS -l nodes=1:ppn=10,walltime=1:00:00

module purge
module load mvapich2/1.6-intel

USER_NAME=user #change to your user name

cd /home/$USER_NAME/lsdyna_job/

mpirun lsdyna i=/home/$USER_NAME/lsdyna_job/billet.forge.209.k
```

- *#PBS -N lsdyna_job*
#PBS -l nodes=1:ppn=10,walltime=1:00:00
Forwarding parameters to the queuing system: task name *lsdyna_job*, number of nodes – 1, number of cores – 10, prescribed time for the task – 1h.
- *module purge*
module load mvapich2/1.6-intel
Deleting default MPI module and downloading *mvapich2/1.6-intel* module.

- `cd /home/$USER_NAME/lldyna_job/`
Change working directory.
- `mpirun lldyna i=/home/$USER_NAME/lldyna_job/billet.forge.209.k`
Starting LS-DYNA program with „*task_name.k*” task.

Adding the task to the queue

Prepared script is added to the queue by a following command:

```
$ qsub lldyna_job.sh
```

As a result of program action, the set of data and log files will be created.

6.5. MATLAB Example

Example can be downloaded from: `/home/_EXAMPLE/matlab_job`.

Script

```
#!/bin/bash

#PBS -N gen_atoms_job
#PBS -l nodes=1:ppn=1,mem=10mb,walltime=00:05:00

USER_NAME=user #change to your user name

cd /home/$USER_NAME/matlab_job

/home/_PROGS/bin/matlab2013a -nodisplay -nodesktop -nojvm -r
"run /home/$USER_NAME/matlab_job/gen_atoms; quit"
```

- `#PBS -N gen_atoms_job`
`#PBS -l nodes=1:ppn=1,mem=10mb,walltime=00:05:00`
Forward parameters to the queuing system: In the given example out task will be executed on one core (*ppn=1*) of a single node (*nodes=1*),

will use 10 MB RAM memory and the task time will not exceed 5 minutes (*walltime=00:05:00*).

- *cd /home/\$USER_NAME/matlab_job*
Going to the directory where the target program will be run. MATLAB *gen_atoms.m* script and *matlab_job.sh* starting script have been earlier copied to the directory.
- */home/_PROGS/bin/matlab2013a -nodisplay -nodesktop -nojvm -r "run /home/\$USER_NAME/matlab_job/gen_atoms; quit"*
Starting MATLAB R2013a in batch mode with „*gen_atoms.m*” script. After completion the program will be closed by *quit* command.

Adding the task to the queue:

Prepared script is added to the queue by a following command:

```
$ qsub matlab_job.sh
```

As the result of program action 3 files will be created:

- *gen_atoms_job.oXXXX* standard program output,
- *gen_atoms_job.eXXXX* program error output,
- *atoms.out* output data.

6.6. Environment module programs

Several programs are available on cluster Grafen as environment modules. If graphical interface is necessary one must use *ssh -X* command to start Grafen session.

The following list presents available program modules:

PROGRAM	STARTING COMMAND
DL_POLY_CLASSIC	<i>module load dl_poly/classic_1.9</i>
PARAVIEW	<i>module load paraview/4.1.0</i>
OPENFOAM	<i>module load openfoam/2.3.0</i>
VMD	<i>module load vmd/1.9.1</i>

Selected applications

Nowadays, numerical simulations have become an important tool to provide insight into all fields of science. It is not surprising to observe increasing demand for development of faster and more complex algorithms solving a wide range of time and length scales, using coupled systems of equations with millions and billions of unknowns. The key component of such simulations is parallel development of high performance computing (HPC) systems, both in terms of hardware and software advances.

Multiscale, multiphysics, great variability, large uncertainties, numerical issues, validation difficulties and extremely complex mathematical models are amongst the common features of contemporary mechanics. There is no doubt that the use of HPC-based simulations became a must for solving scientific and engineering problems. It implies that the new methods and algorithms for mechanical engineering, material science, energy, aerodynamics, and computational biology have to be developed for efficient exploration of modern HPC architectures.

In this chapter we show only few examples of complex computational applications, to illustrate problems where the use of HPC computers appears desirable. With the recent improvement of the IPPT computational environment, we do hope to note further highlights justifying this investment very soon.

7.1. Parameter sensitivity of a tensegrity model of a tissue

Eligiusz Postek

7.1.1. Introduction

The cytoskeleton can be modelled as a tensegrity structure [1]. The role of the cytoskeleton is continuously investigated. It is believed that the mechanical environment is very important to the cells behavior. We cite here, “Change the mechanical stresses on cancer cells and they can start to behave more like healthy ones” [1]. This

is the statement that the mechanical actions affect the living organisms from the level of the entire body to the level of the cell and below. The displacements and stresses are changing in the biological materials continuously due to the growth, division and death of the cells [2, 3].

7.1.2. Methodology, mechanical model

The elementary cell consists of nucleus, actins, microtubules, membrane and collagen. We will deal with the mechanical model of the cytoskeleton. The cytoskeleton consists of actins and microtubules. The actins act as tendons and the microtubules act as compressed struts. They build the tensegrity structure of the cell. The observed behavior of the elementary cells is viscoelastic. The cells are pre-stressed and keep approximately their volume when deformed and they stiffen when undergoing tension. These conditions fulfil the elementary icosahedron based tensegrity structure, Fig. 7.1.1 [4].

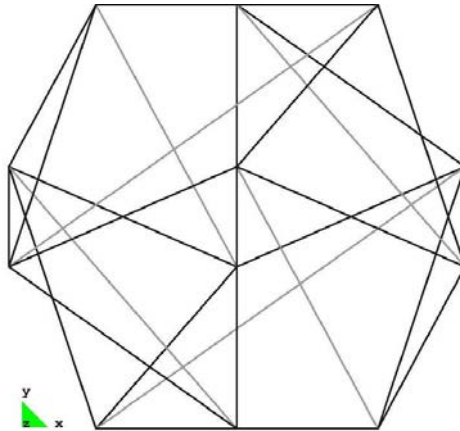


Figure 7.1.1. Elementary tensegrity cell; geometry, struts under compression – black and tendons under tension – grey.

The inherent feature of the model is the parameter sensitivity analysis [5–9]. In particular, the concept of comparison of the behavior of structures by means of the parameter sensitivity analysis was developed in the [5]. The concept was developed for the geometrically nonlinear systems in [7]. The numerical algorithms for the parameter sensitivity analysis were developed in the [8, 9].

The displacements and stresses fields are sensitive to the different parameters, for example, the geometrical parameters like lengths of the actins and microtubules, their cross-sectional areas, constitutive parameters like Young's modulus, relaxation time and sizing parameters. The parameters can be grouped into clusters representing the

cells and groups of clusters. The idea gives information about the influence of cells, their elements and groups of cells. In fact, while observing the derivatives we obtain an additional tool to evaluate the effects of the mechanotransduction. Therefore, we believe that the parameter sensitivity analysis should be a part of the analysis of cell assemblies.

7.1.3. Mathematical formulation

The discrete nonlinear incremental finite element equation of equilibrium fulfilled at the end of the time step after all Newton-Raphson iterations is of the form [8, 9]

$$\left(\int_{\Omega^t} \mathbf{B}^T {}_t^t \bar{\boldsymbol{\tau}} \mathbf{B}^T d\Omega^t \right) \Delta \mathbf{q} + \int_{\Omega^t} \mathbf{B}^T \Delta \mathbf{S} d\Omega^t = \int_{\Omega^t} \mathbf{N}^T \Delta \mathbf{f} d\Omega^t + \int_{\partial\Omega_\sigma^t} \mathbf{N}^T \Delta \mathbf{t} d(\partial\Omega_\sigma^t), \quad (7.1.1)$$

where \mathbf{B} is the linear operator, \mathbf{B}' is the nonlinear operator, \mathbf{N} is the shape functions matrix, ${}_t^t \bar{\boldsymbol{\tau}}$ is the Cauchy stress matrix, $\Delta \mathbf{q}$ is the displacement increment, $\Delta \mathbf{S}$ is the stress increment, $\Delta \mathbf{f}$ is the body forces increment, and $\Delta \mathbf{t}$ is the tractions increment. The integration is over the domain Ω and the boundary $\partial\Omega_\sigma$ where the stress type boundary conditions are fulfilled. The variables are evaluated at the beginning of the step that means at time t (upper t index). The variables are referred to the configuration at time t (lower t index). In the Updated Lagrangian configuration, we look for the variables at time $t+\Delta t$.

We define the parameter sensitivity analysis as an investigation of a general performance functional valid on the body Ω^t and its boundary $\partial\Omega_\sigma^t$ depending on displacements \mathbf{q} , stresses \mathbf{S} and a parameter h . The functional is given as follows:

$${}^{t+\Delta t}_t \Phi = \int_{\Omega^t} {}^{t+\Delta t}_t G(\mathbf{S}, \mathbf{q}, h) d\Omega^t + \int_{\partial\Omega_\sigma^t} {}^{t+\Delta t}_t g(\mathbf{q}, h) d(\partial\Omega_\sigma^t). \quad (7.1.2)$$

The goal is to find the derivative of the functional with respect to the parameter h at time $t+\Delta t$. In other words, we need to find the sensitivity of the functional (7.1.2). We do not consider the shape sensitivity.

Let's assume that the stresses, and the displacements depend on the parameter h . Therefore, we can differentiate the equation of equilibrium with respect to the parameter h as follows:

$$\int_{\Omega^t} \mathbf{B}^T \frac{d\Delta \mathbf{S}}{dh} d\Omega^t + \int_{\Omega^t} \mathbf{B}'^T \frac{d {}_t^t \bar{\boldsymbol{\tau}}}{dh} \mathbf{B}' d\Omega^t \Delta \mathbf{q} + \int_{\Omega^t} \mathbf{B}'^T {}_t^t \bar{\boldsymbol{\tau}} \mathbf{B}' d\Omega^t \frac{d\Delta \mathbf{q}}{dh} = \frac{\partial \Delta \mathbf{Q}}{\partial h}. \quad (7.1.3)$$

The stress increment depends on the total stress \mathbf{S} at time t , strain increment $\Delta\mathbf{e}$, and the parameter h . Therefore, the definition of the constitutive tangent is as follows:

$$\Delta\mathbf{S} = \Delta\mathbf{S}({}^t\mathbf{S}, \Delta\mathbf{e}, h), \quad \frac{\partial\Delta\mathbf{S}}{\partial\Delta\mathbf{e}} = \mathbf{C}^{con}({}^t\mathbf{S}, \Delta\mathbf{e}, h). \quad (7.1.4)$$

Differentiating the stress increment we obtain:

$$\frac{d\Delta\mathbf{S}}{dh} = \frac{\partial\Delta\mathbf{S}}{\partial\Delta\mathbf{e}} \frac{d\Delta\mathbf{e}}{dh} + \frac{\partial\Delta\mathbf{S}}{\partial\mathbf{S}} \frac{d\mathbf{S}}{dh} + \frac{\partial\Delta\mathbf{S}}{\partial h}. \quad (7.1.5)$$

Employing the definition of the constitutive tensor (7.1.4) and the finite element discretization we obtain the following form of the stress increment derivative:

$$\frac{d\Delta\mathbf{S}}{dh} = \mathbf{C}^{con}\mathbf{B} \frac{d\Delta\mathbf{q}}{dh} + \frac{d\Delta\mathbf{S}}{dh} \Big|_{\Delta\mathbf{q}(h)=const}. \quad (7.1.6)$$

The symbol $\Big|_{\Delta\mathbf{q}(h)=const}$ is a consequence of the assumption that the in the Updated Lagrangian approach the reference configuration is not dependent on the parameter h . Substituting the stress derivative increment into the Eq. (7.1.3) we obtain the equation for displacement increment sensitivities:

$$\left(\int_{\Omega'} \mathbf{B}^T \mathbf{C}^{con} \mathbf{B} d\Omega' + \int_{\Omega'} \mathbf{B}'^T {}^t\bar{\boldsymbol{\tau}} \mathbf{B}' d\Omega' \right) \frac{d\Delta\mathbf{q}}{dh} = \frac{d\Delta\mathbf{Q}}{dh} - \left[\int_{\Omega'} \mathbf{B}'^T \frac{d\Delta\mathbf{S}}{dh} d\Omega' \Big|_{\Delta\mathbf{q}(h)=const} + \int_{\Omega'} \mathbf{B}'^T \frac{d'{}^t\bar{\boldsymbol{\tau}}}{dh} \mathbf{B}' d\Omega' \Delta\mathbf{q} \right]. \quad (7.1.7)$$

In short, the sensitivity equation takes the form:

$$\left(\int_{\Omega'} \mathbf{B}'^T {}^t\bar{\boldsymbol{\tau}} \mathbf{B}' d\Omega' \right) \frac{d\Delta\mathbf{q}}{dh} + \int_{\Omega'} \mathbf{B}'^T \Delta\mathbf{S} d\Omega' = \frac{d\Delta\mathbf{Q}}{dh} - \frac{d\Delta\mathbf{R}}{dh} \Big|_{\Delta\mathbf{q}(h)=const}. \quad (7.1.8)$$

On the right hand side (r.h.s.) of the Eq. (7.1.8), we have the sensitivity of the loading increment and the sensitivity of the internal force increment $\Delta\mathbf{R}$. The last derivative consists of the last two terms of the Eq. (7.1.7).

To complete the algorithm, we need to make two steps, firstly, accumulate of the displacement and the stress sensitivities:

$$\frac{d^{t+\Delta t} \mathbf{q}}{dh} = \frac{d^t \mathbf{q}}{dh} + \frac{d\Delta \mathbf{q}}{dh}, \quad \frac{d^{t+\Delta t} \mathbf{S}}{dh} = \frac{d^t \mathbf{S}}{dh} + \frac{d\Delta \mathbf{S}}{dh}, \quad (7.1.9)$$

and secondly, calculate the sensitivity of the performance functional:

$$\begin{aligned} \frac{d^{t+\Delta t} \Phi}{dh} = \int_{\Omega'} \left[\frac{\partial^{t+\Delta t} G}{\partial^t \mathbf{S}} \frac{d^t \mathbf{S}}{dh} + \frac{\partial^{t+\Delta t} G}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dh} + \frac{\partial^{t+\Delta t} G}{\partial h} \right] d\Omega' \\ + \int \frac{\partial^{t+\Delta t} g}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dh} d(\partial \Omega'_\sigma) + \int \frac{\partial^{t+\Delta t} g}{\partial h} d(\partial \Omega'_\sigma). \end{aligned} \quad (7.1.10)$$

The constitutive model is viscous-elastic such as the stress increment depends on total stress \mathbf{S} , the shear modulus (G), the bulk modulus (K) and the strain increment $\Delta \mathbf{E}$ as follows:

$$\Delta \mathbf{S} = \mathbf{C}^{con}(\mathbf{S}, G, K) \Delta \mathbf{E}, \quad G(t) = G_o + \sum_{i=1}^n G_i \exp\left(-\frac{t}{\lambda_i}\right), \quad (7.1.11)$$

where t is the time and λ_i are the relaxation times of the particular parallel dampers. The Eq. (7.1.11) describe the generalized Maxwell model [10].

Considering the truss elements the operators \mathbf{B} and \mathbf{B}' are of the form:

$$\mathbf{B} = \begin{bmatrix} -\frac{1}{l} & 0 & 0 & \frac{1}{l} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B}' = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}, \quad (7.1.12)$$

where l is the length of truss element. The operators (7.1.12) are derived using the linear shape functions matrix, [12].

$$N = \begin{bmatrix} 1-\xi/l & 0 & 0 & \xi/l & 0 & 0 \\ 0 & 1-\xi/l & 0 & 0 & \xi/l & 0 \\ 0 & 0 & 1-\xi/l & 0 & 0 & \xi/l \end{bmatrix}. \quad (7.1.13)$$

The constitutive matrix for the uni-axial stress state is a particular case of the Eqs. (7.1.11) and reads:

$$\Delta S = C^{con}(S, E_o, E_i) \Delta e_{xx}, \quad E(t) = E_o + \sum_{i=1}^n E_i \exp\left(-\frac{t}{\lambda_i}\right), \quad (7.1.14)$$

where Δe_{xx} is the uni-axial strain, E_o is the Young's modulus in the Hookean element, and E_i are the Young's moduli in the parallel Maxwell elements.

7.1.4. Computer implementation

The parallel version of the program includes the solver MULTifrontal Massively Parallel Solver, MUMPS [12, 13]. We use the Newton-Raphson technique for solving the equation of equilibrium (7.1.1). The sensitivity algorithm creates additional right hand sides (7.1.8). We need to solve for as many right hand sides as the design variables. The design sensitivity equation is solved using the triangularized form of the last stiffness obtained in the last iteration loop. The algorithm is summarized in Fig. 7.1.2.

- 1) Solve the Eq. (7.1.1) for displacement increments
- 2) Create the r.h.s vectors of the Eq. (7.1.8), as many as parameters h
- 3) Solve the Eq. (7.1.8) for sensitivity increments
- 4) Calculate the stress increment sensitivity (7.1.6)
- 5) Accumulate the displacement and the stress sensitivities (7.1.9)
- 6) Calculate the sensitivities of the performance functional (7.1.2)
- 7) Pass to the next incremental step

Figure 7.1.2. The outline of the numerical algorithm.

We need to solve for as many right hand sides as the design variables. The design sensitivity equation is solved using the triangularized form of the last stiffness obtained in the last iteration loop.

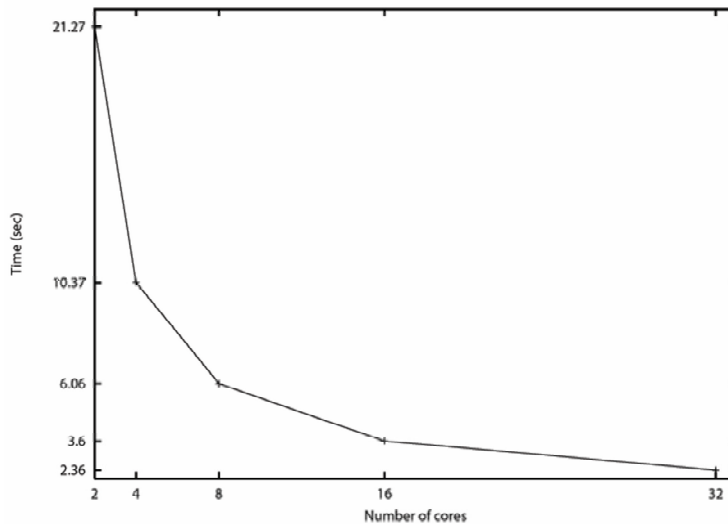


Figure 7.1.3. Speed-up of the calculations.

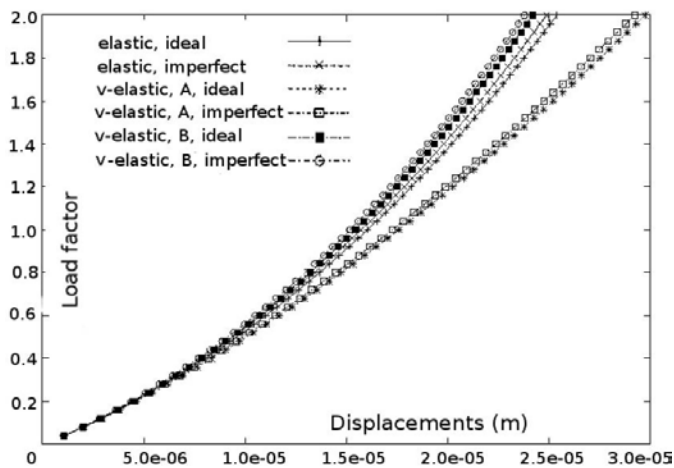


Figure 7.1.4. Stiffening effect in the tensioned cell, load versus displacement.

An example of calculations with the presented algorithm shows the speed-up, Fig. 7.1.3. The 3D system consists of 80400 nodes and 270399 bar elements. The computer time for triangularization of the system of equations for one time step is shown in the Fig. 7.1.4. It is 21.27 sec with 2 cores, and 2.36 sec with 32 cores. The computer is the cluster IPPT “Grafen” that is presented in the sections before. The 3D system is defined in the first example in the paragraph below.

7.1.5. Numerical illustration

Single cell analysis

We deal with the elementary cell model with equivalent actins and microtubules shown in the Fig. 7.1.1. The results presenting the stiffening effect are shown in Fig. 7.1.2. We adopt the following data, namely, height of the cell $64\ \mu\text{m}$, cross-sectional areas of the tendons (filaments) $10\ \text{nm}^2$, cross-sectional areas of the struts (microtubules) $190\ \text{nm}^2$, Young's moduli of the tendons $2.6\ \text{GPa}$ and the struts $1.2\ \text{GPa}$, initial prestressing forces $20\ \text{nN}$, maximum loading $0.1\ \text{N}$, relaxation time $1.0\ \text{sec}$, E_i/E_o ratio 0.91 (case A) and 0.1 (case B). We may note certain effect of the imperfect position of the nodes and higher stiffening effect for higher E_i/E_o ratios (case A).

Single layer of cells analysis

We consider a honeycomb pattern of the single layer cell matrix, Fig. 7.1.3, that is made of the elementary cells presented in Fig. 7.1.1. We assume the same data for each single cell as in the generic cell above. The layer of cells can slide and is fixed on its right side preventing rigid body motions, Fig. 7.1.3. It is loaded with the concentrated forces on the left side. Each concentrate force is $100.0\text{E-}04\ \text{N}$. The tissue is made of 10000 elementary cells. We observe the displacement sensitivities and consider the tissue with a punched hole and a slit. Having the displacement sensitivities, we can evaluate the performance functional at each place of the tissue.

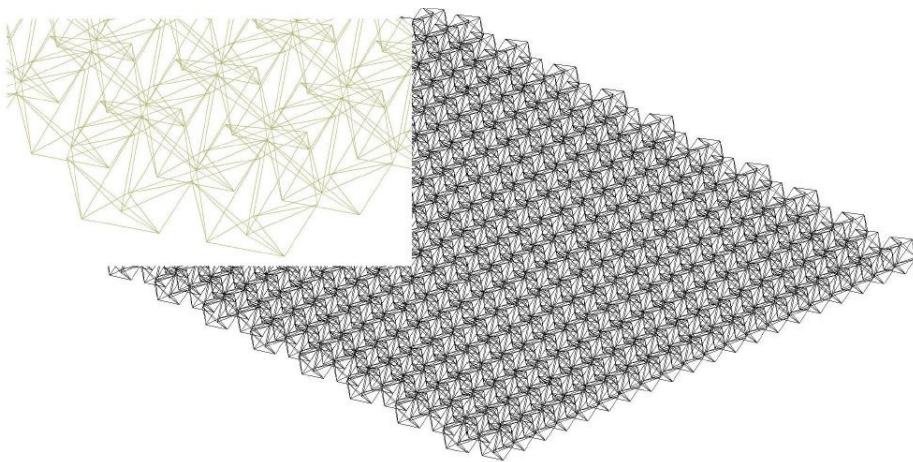


Figure 7.1.5 Matrix of elementary cells: 3D view and a corner detail.

In our case, the performance functional is $\Psi = q$ where q is calculated at nodes of the mesh. In this case, the displacement sensitivity fields are the parameter sensitivities of the functional. The displacement fields are shown in Fig. 7.1.4.

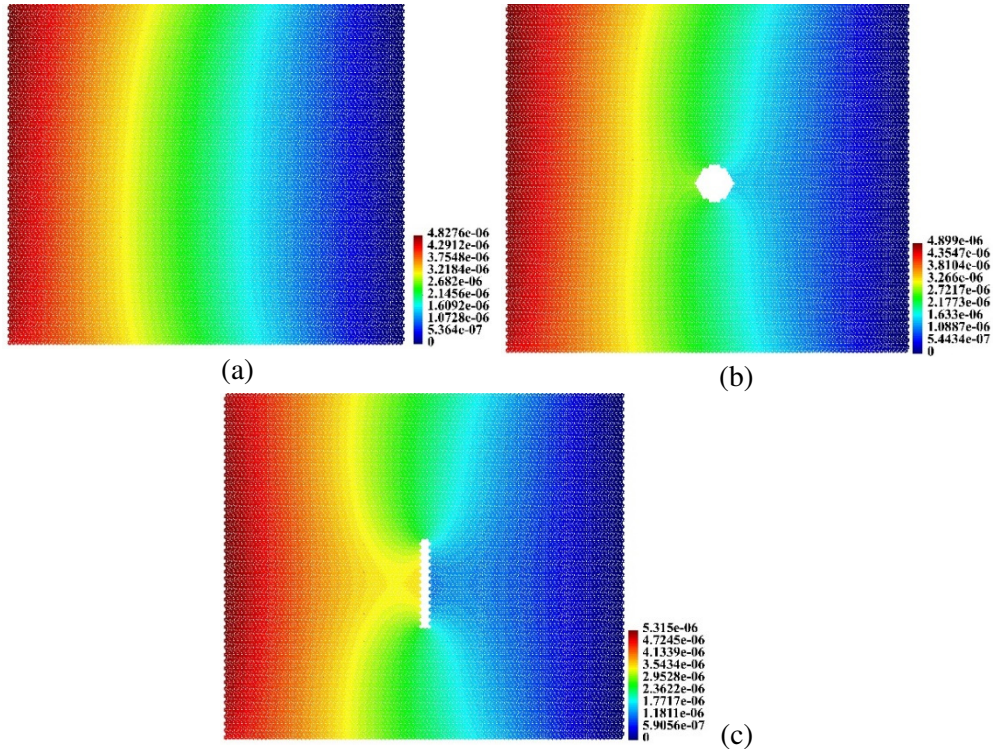


Figure 7.1.6. Displacement fields; (a) ideal tissue, (b) hole, (c) slit.

Specifically, we investigate the displacement sensitivities with respect to the variables defined as the set of the lengths of the microtubules in 4 cells. The positions of the cells are shown in the Fig. 7.1.5. It demonstrates the effect of simultaneous perturbation of the microtubules in 4 cells. Supposing, we would assume simultaneous perturbation of all microtubules in one cell. It would reflect the effect of homogenous growing of one cell. The effect of the growth is evaluated employing the sizing parameters. In this case, we do not consider the shape sensitivity. The cells are easily visible in Fig. 7.1.6 since the design sensitivity gradients are the highest with the cells forming hot spots. When comparing the maximum sensitivities, we note that the less sensitive with respect to the defined set of the design parameters is the ideal system, Fig. 7.1.6a and the most sensitive is the system with the slit, Fig. 7.1.6c. The system with the hole is moderately sensitive, Fig. 7.1.6b.

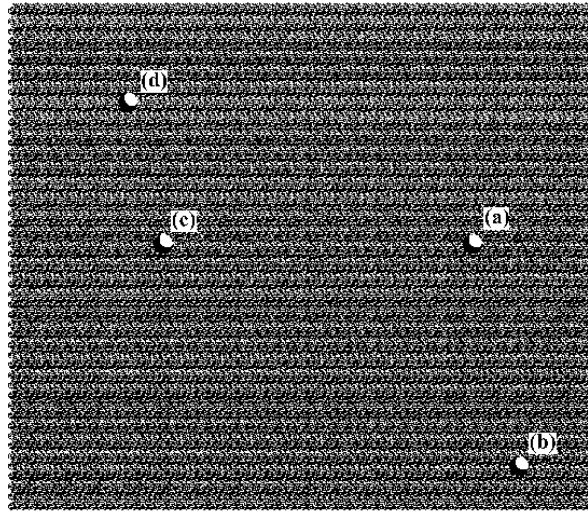


Figure 7.1.7. Positions of the cells that stand for the set of parameters.

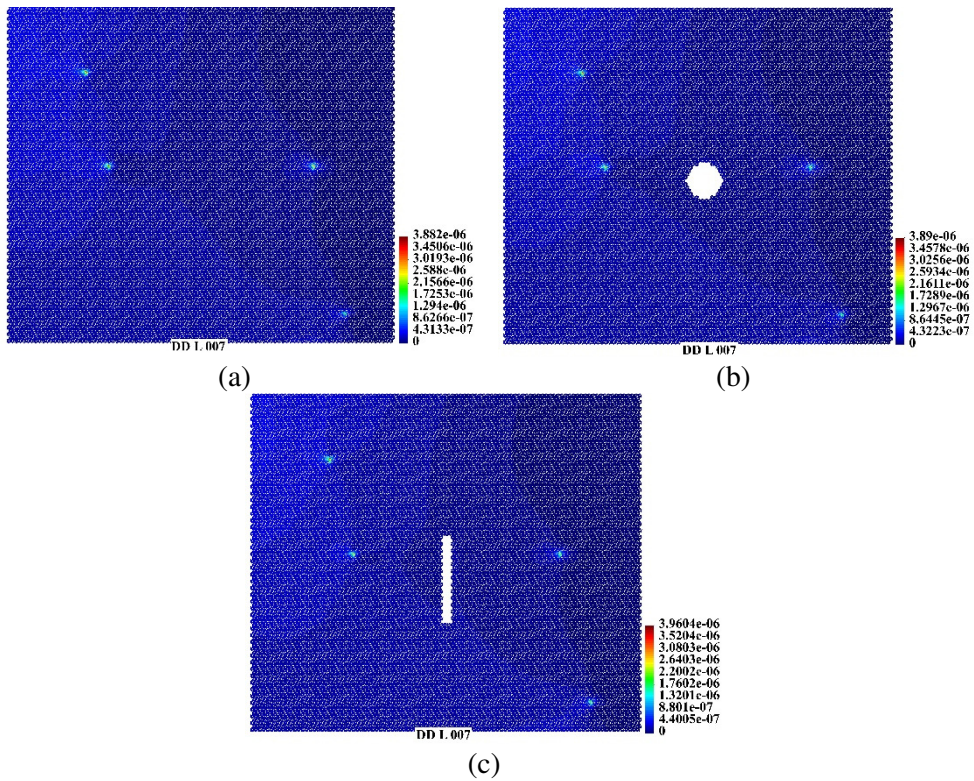


Figure 7.1.8. Parameter sensitivity field: (a) ideal system., (b) hole, (c) slit.

7.1.6. Final remarks

We have presented the implemented design sensitivity algorithm valid for nonlinear path dependent systems. The application is the computational systems biology. We have described numerical aspects of the implementation with parallel multifrontal solver. We have shown a numerical example and a possible scenario of application of the presented theory.

It has been found that the parallel solver MUMPS suits well to the large nonlinear design sensitivity problems because it is possible to solve several right hand sides of the sensitivity equation keeping the triangularized sparse tangent stiffness throughout the process of the elimination of the right hand sides.

We have shown that it is possible to form the design variables into groups reflecting the growth of single cells and the growth of group of cells. The effects of single cell growth on the displacement fields in the tissue was shown in [14].

References

- [1.] C. Ainsworth. Stretching the imagination. *Nature*, 456:696-699, 2008.
- [2.] E. Postek, R. Smallwood, and R. Hose. Nodal positions displacement sensitivity of an elementary icosahedron tensegrity structure. *X Int. Conference on Computational Plasticity, COMPLAS X, CD paper*. Barcelona: Spain, 2009.
- [3.] E. Postek, Concept of an Agent-stress Model of a Tissue. *Technische Mechanik*, 32 (2-5):518-529, 2012.
- [4.] D. Stamenovic. Effects of cytoskeletal prestress on cell rheological behaviour, *Acta Biomaterialia*, 1 (3):255-262, 2005.
- [5.] E. Postek and T. D. Hien. On the application of the sensitivity analysis to the design of spatial bar structures. *Archives of Civil Engineering*. 37 (1-2), 1992.
- [6.] M. Kleiber. Shape and non-shape structural sensitivity analysis for problems with any material and kinematic non-linearity. *Computer Methods in Applied Mechanics and Engineering*, 108 (1-2):73-97, 1993.
- [7.] M. Kleiber, T.D. Hien, and E. Postek. Incremental finite element sensitivity analysis for nonlinear mechanics applications. *International Journal for Numerical Methods in Engineering*. 37(19):3291-3308, 1994.
- [8.] E. Postek. Numerical Parameter Design Sensitivity of Large Nonlinear Structural Systems. *PhD Thesis* (in Polish), IPPT PAN, 1997.
- [9.] M. Kleiber, H. Antunez, T. D. Hien and P. Kowalczyk. *Parameter Sensitivity in Nonlinear Mechanics: Theory and Finite Elements*. Wiley & Sons, 1997.
- [10.] W. Nowacki. *Teoria pełzania*. Arkady, 1963.
- [11.] M. Kleiber and Cz. Woźniak. *Nonlinear Mechanics of Structures*. Polish Scientific Publishers. Kluwer Academic Publishers, 1991.

- [12.]P.R. Amestoy, I.S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal of Matrix Analysis and Applications*, 23 (1):15–41, 2001.
- [13.]P.R. Amestoy, A. Guermouche, J.Y. L'Excellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32 (2):136–156, 2006.
- [14.]E. Postek, R. Smallwood, and R. Hose. Behaviour of tensegrity cells assembly during single cell growth. *Conference Proceedings CMBE09, 1st International Conference on Mathematical and Computational Biomedical Engineering, June 29–1 July, Swansea, UK, 415–418, 2009.*

7.2. Generic models of linear and non-linear visco-elastic surface deformation above a fault

Eligiusz Postek

7.2.1. Problem statement

The element of the seismic mic slip. It appears at the moment of the earthquake. For example, analyses of such a slip are presented in the paper [1, 2]. The analyses are done with the InSAR. An example of such a InSAR analysis is shown in Fig. 7.2.1. This a deformation pattern around a strike-slip fault in Tibet after the M_w 7.6 earthquake that occurred in 1997. High efforts are made to predict the post-seismic deformation that is caused by the co-seismic slip. However, the efforts are still not very successful. It is believed that the problem is still lack of sufficient data on the faults and the distribution of the slip. The problem is that the mechanism of a triggering of an earthquake is not well as yet.

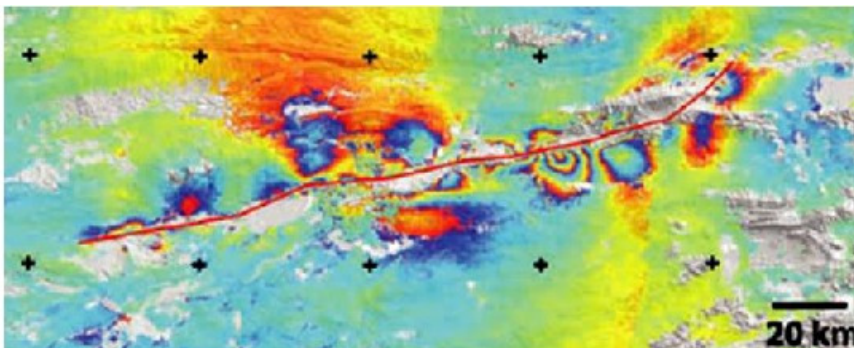


Figure 7.2.1. Manyi (Tibet) fault, the InSAR pattern of the deformation after the earthquake (with the Prof. Tim Wright's permission, University of Leeds).

We present a class of viscoelastic constitutive models [3]. This is a family of non-linear Maxwell models (non-Newtonian). The generic model of a fault is presented. The constitutive law depends on the temperature (Arrhenius law), time and strain hardening. The program is aimed at calculations of layered and arbitrarily variable systems for co-seismic, post-seismic slip and rebounds. The evidence of applicability of such models is shown in [4].

The numerical example that is described in the last section is aimed at the a qualitative analysis of the strike-slip fault.

7.2.2. Nonlinear creep

The problem of the non-linear creep is defined with the small strains assumption and additivity of the elastic and creep strain rates:

$$\dot{\epsilon} = \dot{\epsilon}^e + \dot{\epsilon}^{cr}. \quad (7.2.1)$$

The total creep strain is a function of stress, temperature and time. The dependence of this parameters is separable:

$$\epsilon^{cr} = F(\sigma, T, t), \quad \epsilon^{cr} = f_1(\sigma)f_2(t)f_3(T). \quad (7.2.2)$$

The temperature function depends on the creep activation energy H, temperature T and the universal gas constant R:

$$f_3(T) = \exp\left(\frac{H}{RT}\right). \quad (7.2.3)$$

We deal with the creep law where stress is in exponent n and time is in exponent m. We have to calculate the creep strain rate. We assume time and strain hardening rules. The strain hardening means that the creep strain rate depends on the total creep strain. The total creep strain is calculated directly from the creep law and substituted into the expression for the creep strain rate. The equations read:

$$\begin{aligned} \epsilon^{cr} &= A \sigma^n t^m f_3(T), \\ \dot{\epsilon}^{cr} &= mA \sigma^n t^{m-1}, \\ \dot{\epsilon}^{cr} &= mA^{\frac{1}{n}} \sigma^{\frac{n}{m}} (\epsilon^{cr})^{\frac{m-1}{m}}. \end{aligned} \quad (7.2.4)$$

To get the relations for the 3D stress state we need to calculate the volumetric and the deviatoric parts of the stress tensor:

$$\begin{aligned}\sigma_m &= \frac{1}{3} \sigma_{kk}, \\ S_{ij} &= \sigma_{ij} - \frac{1}{3} \delta_{ij} \sigma_{kk}.\end{aligned}\quad (7.2.5)$$

The equivalent stress and strain read:

$$\begin{aligned}\sigma_{eff} &= \frac{1}{\sqrt{2}} \left[(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{xx} - \sigma_{zz})^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{xz}^2) \right]^{\frac{1}{2}}, \\ \varepsilon_{eff}^{cr} &= \frac{\sqrt{2}}{3} \left[(\varepsilon_{xx} - \varepsilon_{yy})^2 + (\varepsilon_{yy} - \varepsilon_{zz})^2 + (\varepsilon_{xx} - \varepsilon_{zz})^2 + 6(\gamma_{xy}^2 + \gamma_{xz}^2 + \gamma_{yz}^2) \right]^{\frac{1}{2}}.\end{aligned}\quad (7.2.6)$$

We will assume that the creep strain develops only during shear deformation. To calculate the creep strain rate we introduce the creep potential using the Prandtl-Reuss condition. The flow function is the Huber-Mises function where λ is determined by substituting into the expression for the equivalent stress and utilizing the definition of the effective creep strain:

$$\dot{\varepsilon}_{ij}^{cr} = \frac{d\varepsilon_{ij}^{cr}}{dt} = \lambda S_{ij}, \quad (7.2.7)$$

where

$$\lambda = \frac{3}{2} mA (\sigma_{eff})^{n-1} t^{m-1}. \quad (7.2.8)$$

The formulae for the creep strain rate, time and strain hardening cases are as follows:

$$\begin{aligned}\dot{\varepsilon}_{ij}^{cr} &= \frac{3}{2} mA (\sigma_{eff})^{n-1} S_{ij} t^{m-1}, \\ \dot{\varepsilon}_{ij}^{cr} &= \frac{3}{2} mA^{\frac{1}{m}} (\sigma_{eff})^{\frac{n-m}{m}} S_{ij} (\varepsilon_{eff}^{cr})^{\frac{m-1}{m}}.\end{aligned}\quad (7.2.9)$$

Additionally, the creep strain rates above are multiplied by the function of temperature which has been omitted for brevity, so far. We will use finite element displacement model. In the results below we assume $m = 1$ in all cases.

7.2.3. Numerical algorithm

We will use finite element displacement model with the explicit time marching scheme [5]. In this particular case, the 3D 10 nodes finite elements are used [6].

1. Calculate the initial solution (elastic):

$$\begin{aligned} \mathbf{K}_e \Delta \mathbf{q} &= \Delta \mathbf{Q}, \\ \mathbf{K}_e &= \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega, \\ \boldsymbol{\sigma} &= \Delta \mathbf{B} \Delta \mathbf{q}, \end{aligned} \quad (7.2.10)$$

where \mathbf{K}_e is the stiffness matrix, $\Delta \mathbf{q}$ is the displacement increment vector, $\Delta \mathbf{Q}$ is the loading vector, \mathbf{D} is the elasticity matrix, \mathbf{B} is the discretized linear operator that acts on the nodal displacements. The continuous form of the \mathbf{B} operator is as follows:

$$\mathbf{A} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{bmatrix}. \quad (7.2.11)$$

2. Calculate the initial time step:

$$\Delta t = \frac{4(2+\nu)}{3EA n \sigma_{eff}^{n-1} f_3(T)}, \quad (7.2.12)$$

where E is the Young's modulus, ν is the Poisson's ratio and f_{fact} is the time step safety factor.

3. Start time integration loop

- a) calculate the stress deviator,
- b) calculate the creep strain increment:

$$\Delta \boldsymbol{\varepsilon}_{cr} = \frac{3}{2} m A \boldsymbol{\sigma}_{eff}^{n-1} \mathbf{S}_{dev} f_3 \left(T \left[(t + \Delta t)^m - t^m \right] \right), \quad (7.2.13)$$

- c) calculate the creep stress increment and the increment of the equivalent internal forces vector:

$$\begin{aligned} \Delta \boldsymbol{\sigma}^{cr} &= \mathbf{D} \Delta \boldsymbol{\varepsilon}^{cr}, \\ \Delta \mathbf{F} &= - \int_{\Omega} \mathbf{B}^T \Delta \boldsymbol{\sigma}^{cr} d\Omega, \end{aligned} \quad (7.2.14)$$

- d) calculate the displacement and strain increments:

$$\begin{aligned} \mathbf{K}_e \Delta \mathbf{q} &= \Delta \mathbf{F}, \\ \Delta \boldsymbol{\varepsilon} &= \mathbf{B} \Delta \mathbf{q}, \end{aligned} \quad (7.2.15)$$

- e) calculate the strain and stress increments:

$$\Delta \boldsymbol{\varepsilon}_i = \Delta \boldsymbol{\varepsilon} - \Delta \boldsymbol{\varepsilon}_{cr}. \quad (7.2.16)$$

The critical time step can be checked for every time increment or the number of checks can be assigned by the user.

7.2.4. Numerical example

We investigate a model of a fault between two crustal blocks. Because of relative movement of the crustal blocks in the contact zones patches of load (tractions) appear. This model imposes such tractions on a single block. This action causes deformation on the surface. The deformation can be monitored by observations of the GPS points or InSAR analysis. The scheme of the crustal block is given in the Fig. 7.2.2.

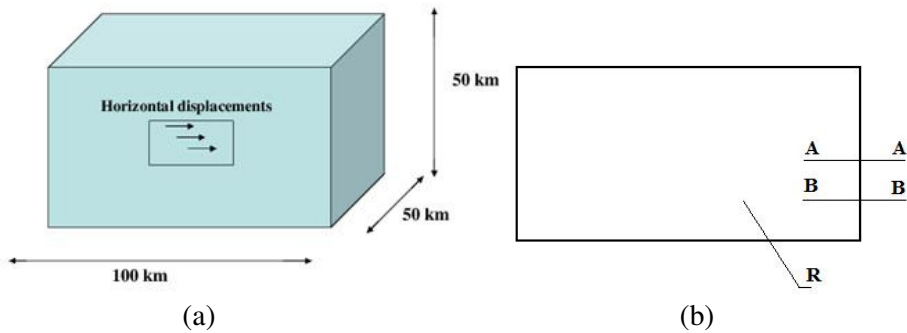


Figure 7.2.2. Crustal block with strike-slip fault (a) analysed cross-sections and point R (b).

The conceptual model is further analysed with a rectangular patch (30 by 10 km) of sinusoidal displacements (co-seismic slip, maximum 5 m), Fig. 7.2.2a. The fault surface is fixed (locked) except for the patch where the displacements are applied. The centroids of the patches are placed at the depths -5 km, -10 km, -15 km, -25 km, -35 km and -45 km, respectively. The Young's modulus is $5.0E+10$ Pa, Poisson's ratio is 0.25 and viscosity coefficient A is $2.2E-20$ $1/(Pa \cdot sec)$. We observe the vertical displacement variation in time of a characteristic point R (position of the point is $x = 8.33$ km, $z = 61.7$ km), Fig. 7.2.2b. We note the qualitative difference in the displacement variation for different n . The displacements tend to steady state solutions almost immediately for $n = 13$ (log time = 0.1), slower for $n = 2$ (log time = 6) and very slow for the linear Maxwell model, $n = 1$ (log time = 10).

We investigate two cross-sections along the block: A-A, $x = 8.33$ km and B-B, $x = 25$ km, Fig. 7.2.3b. The elastic displacements are shown in the Fig. 7.2.3.

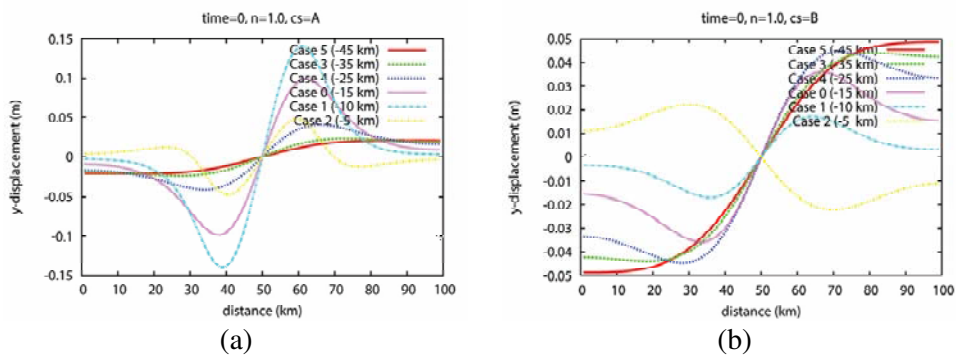


Figure 7.2.3. Elastic displacements; a) cross-section A-A, b) cross-section B-B.

Depending on the position of the patch the maximum displacements vary from 0.14 m to 0.02 m in the cross-section A-A and from 0.05 m to 0.02 m in the cross-section B-B. They are the largest in the cross-section A-A for the shallower co-seismic slips which are close to the fault. However, the displacements in the cross-section B-B are larger for the deeper co-seismic slips than for the shallower. The elastic vertical displacements pattern for co-seismic slip -15 km is shown on in Fig. 7.2.5a. The vertical displacements with the cross-sections A-A and B-B at $\log(\text{time})$ equal to 10, 6 and 1 are given in the Fig. 7.2.3. The displacements in the cross-sections A-A are negligible for deep slips for $n = 2$ and $n = 3$, what is different from the situation when $n = 1$. The displacements did not decay. Considering the cross-section B-B we note that the displacements are decaying for non-Newtonian laws. The vertical displacements patterns for the case -15 km are shown in Fig. 7.2.5 b, c, d. The largest vertical displacements are for the linear Maxwell model. The lowest and the most concentrated are for the non-newtonian model with $n = 3$. The model consists of 46080 10-nodes tetrahedral elements and 100800 nodes.

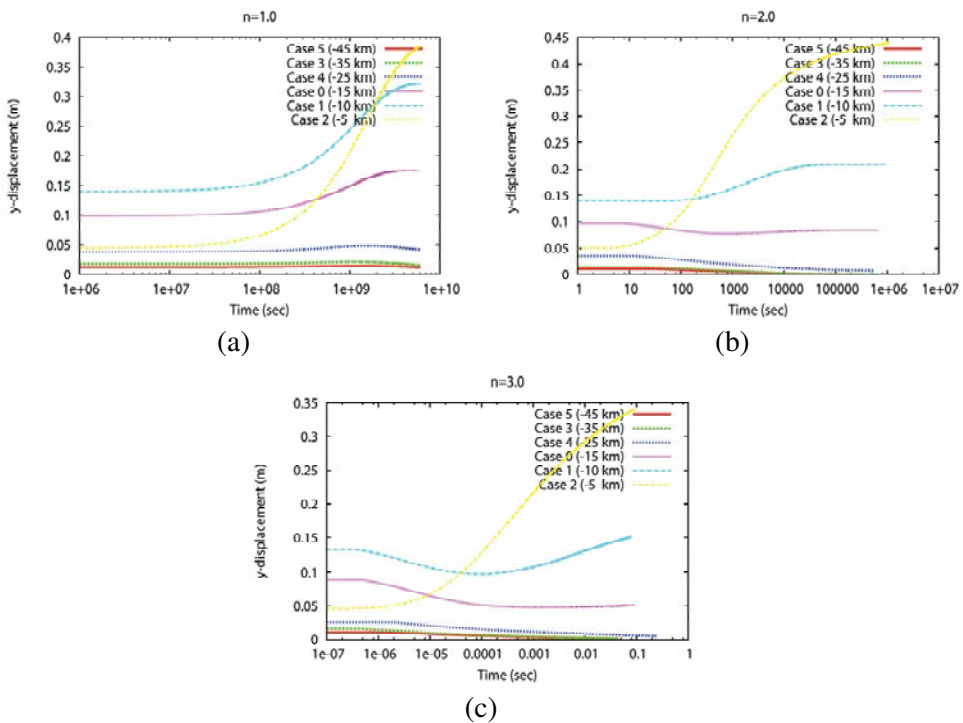


Figure 7.2.4. Dependence of controlled displacement on time for the exponent that equals 1 (a), 2 (b) and 3 (c).

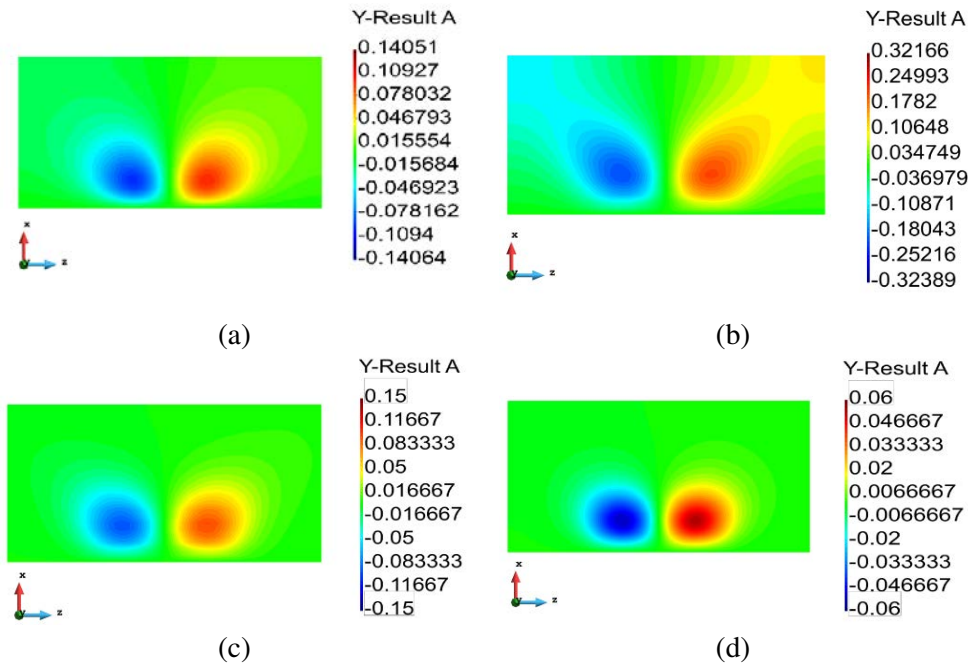


Figure 7.2.5. Vertical displacements distribution, elastic solution (a), displacements along the cross-section A-A (a) and the cross-section B-B.

7.2.5. Final remarks

We have presented a generic model of the fault using newly developed version of Oregano program, for example [7]. The new version is called Oregano-VE which allows general Maxwell viscoelastic deformation fields to be computed [8, 9]. We verify the program solving uni-axial stress problem for linear and nonlinear creep deformation. Our 3D example problem shows displacement fields on the surface of a 3D crustal block. We examine the effect of traction forces and, first of all, the application of a step-function displacement to a patch of the block, representing a slipped fault surface. We observe the co-seismic deformation patterns and the post-seismic creep. We identify the qualitative differences between Newtonian and non-Newtonian behaviour of the block. The presented algorithm can be extended immediately to other nonlinear creep laws [10, 11].

References

- [1.] G. J. Funning, B. Parsons, and T. J. Wright. Fault slip in the 1997 Manyi, Tibet earthquake from linear elastic modelling of InSAR displacements. *Geophysics J. Int.*, 169:988–1008, 2007.

- [2.] S. Jonsson, H. Zebker, P. Segall, and F. Amelung. Fault Slip Distribution of the 1999 Mw 7.1 Hector Mine, California, Earthquake, Estimated from Satellite Radar and GPS Measurements. *Bulletin of the Seismological Society of America*, 92 (4):1377–1389, 2002.
- [3.] R. K. Penny and D. L. Marriott. *Design for Creep*. McGraw Hill, 1971.
- [4.] A. M. Freed and R. Bürgmann. Evidence of power-law flow in the Mojave desert mantle. *Nature* 430:548–551, 2004.
- [5.] M. Kojic and K. J. Bathe. *Inelastic Analysis of Solids and Structures*. Springer, 2005.
- [6.] M. Kleiber and Cz. Woźniak. *Nonlinear Mechanics of Structures*. Polish Scientific Publishers, Kluwer Academic Publishers, 1991.
- [7.] P. Molnar, G. A. Houseman. Rayleigh-Taylor instability, lithospheric dynamics, surface topography at convergent mountain belts, and gravity anomalies. *Journal of Geophysical Research – Solid Earth*, 118:2544–2557, 2013.
- [8.] E. Postek, G. A. Houseman, and P.K. Jimack. Generic Models of Linear and Non-linear Visco-elastic Surface Deformation above a Fault, *EGU2008-A-04783, GD8-1TH2P-0093, Session GD8: Modelling and Monitoring the Deformation and State of Stress of the Lithosphere (co-sponsored by the International Lithosphere Program Task Force VII)*. *European Geophysical Union General Assembly*. Vienna, 13–18 April, 2008.
- [9.] E. Postek, G.A. Houseman, and P. K. Jimack. Development of the program OREGANO-V, towards non-linear visco-elasticity. *Centre for the Observation and Modelling of Earthquakes and Tectonics, COMET meeting at Oxford University*. January 8th, 2008.
- [10.] H.J. Frost and M. F. Ashby. *Deformation-Mechanism Maps. The Plasticity and Creep of Metals and Ceramics*, Pergamon Press, 1982.
- [11.] H. Sone and M. D. Zoback. Mechanical properties of shale-gas reservoir rocks, Part 2: Ductile creep, brittle strength, and their relation to the elastic modulus. *Geophysics*, 78 (5):D393–D402, 2013.

7.3. Microstructure-based modelling of sound absorption in rigid porous media

Tomasz Zieliński

Sound propagation and absorption in porous media with a rigid frame (skeleton) are usually modelled using the fluid-equivalent approach, where the porous material is substituted by an effective dispersive fluid. The Johnson-Champoux-Allard-Lafarge semi-phenomenological model and its variations are perhaps the most efficient and frequently used realizations of that idea. These models use from six to eight parameters that, in fact, results purely from the geometry of porous microstructure; they are, i.e.: the total open porosity, the tortuosity, the (viscous) permeability and its thermal analogue, the viscous and thermal characteristic lengths, etc. Other necessary

parameters are some well-known physical properties of the fluid in pores, namely: the density, the viscosity, the Prandtl number, the ratio of specific heats, and the ambient mean pressure.

The multiscale modelling relies on the fact that the geometrical (or transport) parameters of a porous medium can be computed from the microstructure by solving some relevant physical problems. In practice, this requires rather large computational resources since the problems must be solved on rather complex realistic domains of porous media using the Finite Element Method. COMSOL Multiphysics software working on the „Grafen” supercomputing cluster allowed to solve the scaled problems of viscous flow (i.e., the Stokes flow), heat transfer, and potential flow (i.e., the Laplace problem) on various Representative Volume Elements for porous materials with oval (spherical) pores of various size filled with air (see Fig. 7.3.1), or for porous media composed from rigid spherical beads of identical diameter where the fluid (the air) fills the open space between the spheres (see Fig. 7.3.2 and 7.3.3). Moreover, some corresponding harmonic problems (namely, the scaled harmonic viscous flow and the scaled thermal flow) were also solved using COMSOL Multiphysics on the „Grafen” supercomputing cluster for many computational frequencies in a wide frequency range, in order to calculate directly from the microstructure some frequency-dependent characteristics relevant for viscous and thermal effects of sound waves dissipation.

The procedures for the direct or hybrid microstructure-based approach for multiscale modelling of sound propagation and absorption in rigid porous media were developed in MATLAB integrated with COMSOL Multiphysics through LiveLink for MATLAB, and were tested using the „Grafen” supercomputing cluster, and some of the results were validated experimentally. For example, Fig. 7.3.2 presents the electric potential field found by solving the Laplace problem defined on the finite element mesh of a fluid domain of the Representative Volume Element constructed for a porous medium of freely-packed rigid spheres. In that case the so-called Simple Cubic (SC) arrangement of spheres was used to create the representative cell; however, the spheres were allowed to overlap slightly in order to adjust to the actual (i.e., measured) porosity of the medium that was 42% (see Fig. 7.3.2). The Finite Element Analysis was carried out and its result served to calculate the average parameters of tortuosity and viscous characteristic length. The analyses of viscous, incompressible flow and thermal flow were also effectuated using this SC-RVE of a porous medium, which allowed to calculate the remaining parameters, like for example, viscous and thermal permeabilities. Similar analyses were also computed for another RVE based on the so-called Body Centered Cubic (BCC) arrangement of spheres. This time, however, the spheres were slightly shifted apart in order to fit the actual porosity of 42%. Fig. 7.3.3 shows the BCC arrangement, finite element mesh of the RVE of the fluid

domain, as well as an exemplary solution of the scaled thermal flow problem, which served to calculate the averaged parameter of the thermal permeability. The macroscopic transport parameters calculated from the relevant microstructural analyses based on the SC and BCC Representative Volume Elements were used by the MATLAB procedures in order to calculate some harmonic characteristics of the considered porous medium, and then the frequency-dependent complex effective speed of sound. Eventually, the surface acoustic impedance and absorption coefficient were calculated for the 106 mm-thick layer of such medium composed of identical rigid spheres with diameter 5.9 mm. All calculations were done for both RVEs and finally the results of multiscale modelling were compared with the result found experimentally from the measurements of such layer carried out in the impedance tube: as it can be seen from Fig. 7.3.4, a rather good agreement was found.

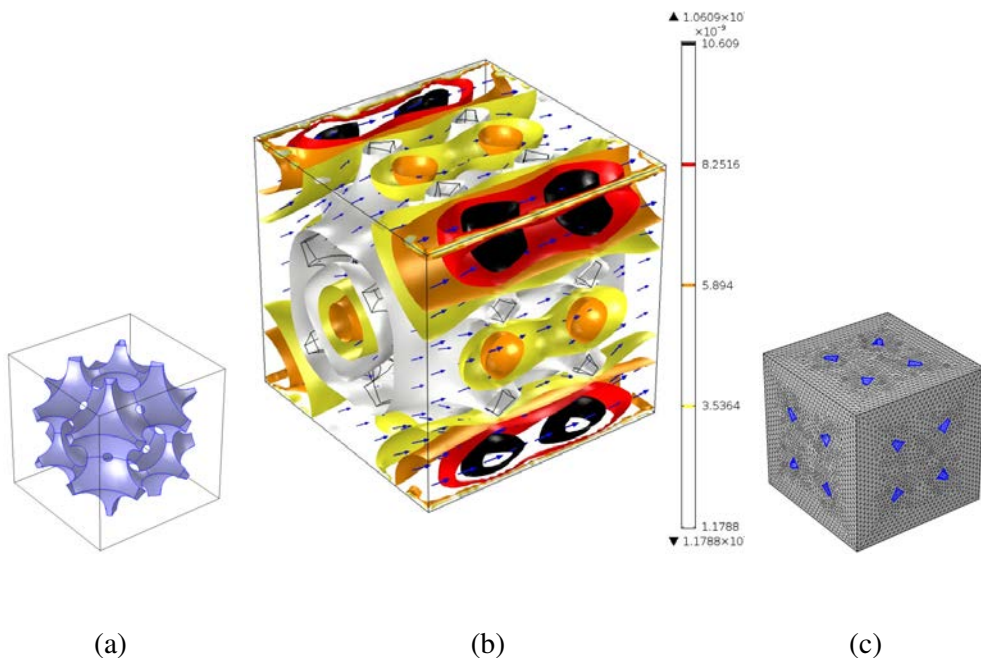


Figure 7.3.1. Periodic RVE showing solid skeleton (a), the corresponding finite element mesh of fluid domain (c), and the viscous permeability field resulting from the scaled solution of the viscous flow through porous medium (b).

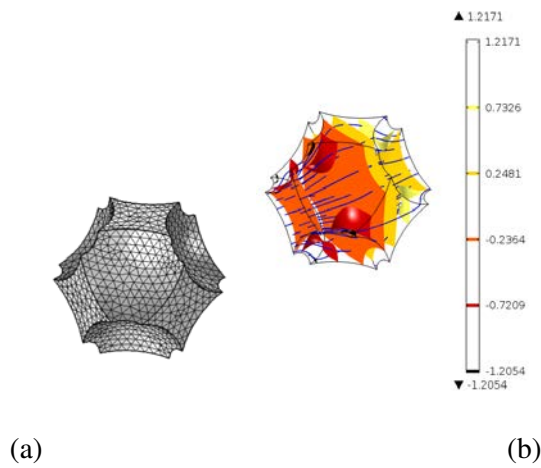


Figure 7.3.2. Periodic RVE with the SC sphere packing, the finite element mesh of fluid domain (a), and the electric potential field resulting from the scaled solution of the Laplace problem inside porous medium (b).

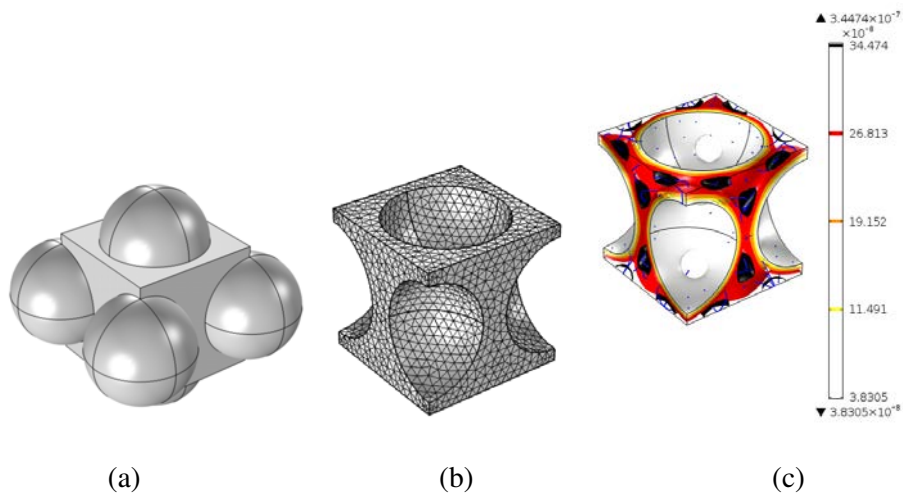


Figure 7.3.3. Periodic RVE with the BCC sphere packing (a), the finite element mesh of fluid domain (b), and the thermal permeability field (c) resulting from the scaled solution of the thermal flow inside porous medium.

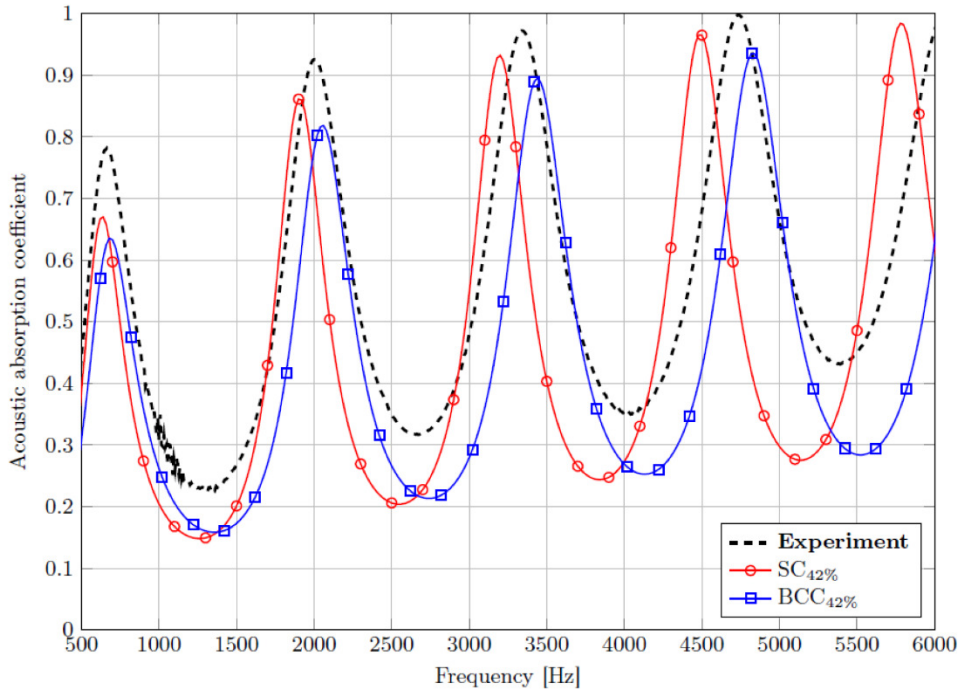


Figure 7.3.4. Acoustic absorption of 106 mm-thick porous layer of identical spheres with diameter of 5.9 mm.

References

- [1.] T. G. Zieliński. Inverse identification and microscopic estimation of parameters for models of sound absorption in porous ceramics. *Proceedings of ISMA 2012: International Conference on Noise and Vibration Engineering*, 17–19 September, Leuven, Belgium, 95–107, 2012.
- [2.] T. G. Zieliński. Representative volume elements, microstructural calculation and macroscopic inverse identification of parameters for modelling sound propagation in rigid porous materials. *Proceedings of ICSV20: 20th International Congress on Sound and Vibration*, 7–11 July 2013, Bangkok, Thailand.
- [3.] T. G. Zieliński. Concurrence of the micro-scale calculation and inverse identification of parameters used for modelling acoustics of porous media. *Proceedings of SolMech 2012: 38th Solid Mechanics Conference*, 27–31 August 2012, Warsaw, Poland.
- [4.] T. G. Zieliński. Multiphysics modelling of sound Absorption in rigid porous media based on periodic representations of their microstructural geometry. *COMSOL Conference 2013*, 23–25 October 2013, Rotterdam, The Netherlands.

- [5.] T. G. Zieliński. Sound absorption of porous layers of loosely-packed rigid spheres: multiscale modelling and experimental validation. *Forum Acusticum* 2014, 7–12 September 2014, Kraków, Poland.
- [6.] T. G. Zieliński. Microstructure-based calculations and experimental results for sound absorbing porous layers of randomly packed rigid spherical beads. *Journal of Applied Physics*, 116(3,Art.034905)1–17, 2014.

7.4. Reliability analysis of reinforced concrete structures

Eligiusz Postek

7.4.1. Introduction

Recently, a construction of new nuclear power plant is planned. Therefore, we decided to remind one of methodologies that is used during design of such plant for failure pressure.

The application of the reliability analysis and reliability based design is very broad. It is particularly useful in the case of taking into account different types of material and geometric imperfections. However, the researchers and the designers are often discouraged with some disadvantages of the stochastic approach which is very often linked with the crude Monte-Carlo method, for example [1]. An improvement of the method is the adaptive Monte-Carlo method. A significant improvement of the methodology is the sensitivity based approach [2-7, 18]. It is particularly useful under the condition of obtaining the necessary design derivatives of the failure functions analytically making the solution independent of the chosen perturbation.

7.4.2. Reliability algorithm

Consider a stochastic process as a vector of random variables \mathbf{U} . The specific realization is the vector $\mathbf{x} = \{\mathbf{x}^\mu, \mathbf{x}^\sigma\}$ where \mathbf{x}^μ and \mathbf{x}^σ are the vectors of the mean values and standard deviations, respectively. This is valid with the assumption of the proper stochastic distributions. The following failure function is defined:

$$g(\mathbf{x}) = 0. \quad (7.4.1)$$

The vector \mathbf{x} is transformed to the system of standard variables \mathbf{u} with the Rosenblatt transformation $\mathbf{x} = \mathbf{T}(\mathbf{u})$. The definition of the reliability index takes the form [5]:

$$\beta = \|\mathbf{u}^*\| = \min \|\mathbf{u}\| \quad \text{with the condition} \quad g(\mathbf{u}) = 0. \quad (7.4.2)$$

A range of methods of calculation of the reliability index exists. For example, Rackwitz-Fiessler method [11]. The failure function g in (7.4.1) depends on the stochastic parameters vector $\mathbf{x}(\mathbf{u})$ and the performance function Ψ . The performance functions depends on displacements vector \mathbf{q} or stress vector \mathbf{S} . Both; the displacements and stresses depend on \mathbf{x} , as well. Therefore, the failure function is of the form:

$$g\{\mathbf{x}(\mathbf{u}), \psi[\mathbf{x}(\mathbf{u})]\} = 0. \quad (7.4.3)$$

The first derivative of the failure function (7.4.3) with respect to standard variables vector takes the form:

$$\frac{\partial g}{\partial \mathbf{u}} = \frac{\partial g}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{u}} + \frac{\partial g}{\partial \psi} \frac{\partial \psi}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{u}}. \quad (7.4.4)$$

The first derivative of the failure function is necessary while performing the optimization procedure (7.4.1). The derivatives $\partial g/\partial \Psi$ and $\partial g/\partial \mathbf{u}$ are usually calculated explicitly. The derivatives $\partial \mathbf{x}/\partial \mathbf{u}$ are obtained exploiting the transformation $\mathbf{x} = \mathbf{T}(\mathbf{u})$. The derivatives $\partial \Psi/\partial \mathbf{x}$ are the sensitivities. They can be calculated using the methods described in [3, 4, 14].

7.4.3. Performance functions

Let us consider a function determined in the domain Ω and its boundary $\partial\Omega_\sigma$ fulfilling the displacement and stress boundary conditions. The function depends on stresses \mathbf{S} and nodal displacements \mathbf{q} . The stresses and the displacements are dependent implicitly on the parameter h and are determined at time $t+\Delta t$ (i.e. at the end of the step) what is consistent with the Updated Lagrangian approach [24, 25]. The function takes the form:

$${}^{t+\Delta t}_t \psi = \int_{\Omega'} {}^{t+\Delta t}_t G(\mathbf{S}, \mathbf{q}, h) d\Omega' + \int_{\partial\Omega'_\sigma} {}^{t+\Delta t}_t g(\mathbf{q}, h) d(\partial\Omega'_\sigma). \quad (7.4.5)$$

The goal of the analysis is to calculate the total derivative of the function (7.4.5) with respect to parameter h . The derivative reads:

$$\begin{aligned} \frac{d^{t+\Delta t} \psi}{dh} = \int_{\Omega'} \left[\frac{\partial^{t+\Delta t} G}{\partial^t \mathbf{S}} \frac{d^t \mathbf{S}}{dh} + \frac{\partial^{t+\Delta t} G}{\partial \mathbf{q}} \frac{d \mathbf{q}}{dh} + \frac{\partial^{t+\Delta t} G}{\partial h} \right] d\Omega^t \\ + \int_{\partial \Omega'_\sigma} \frac{\partial^{t+\Delta t} g}{\partial \mathbf{q}} \frac{d \mathbf{q}}{dh} d(\partial \Omega'_\sigma) + \int_{\partial \Omega'_\sigma} \frac{\partial^{t+\Delta t} g}{\partial h} d(\partial \Omega'_\sigma). \end{aligned} \quad (7.4.6)$$

The particular case of the function (7.4.5) is the displacement constraint condition

$$\psi = |q| - q^a < 0, \quad (7.4.7)$$

where q is a chosen displacement and q^a is its allowable value.

7.4.4. Parameter sensitivity

Direct differentiation method (DDM)

The goal is to obtain the increment of the sensitivity of the function (7.4.6) with respect to the parameter h . The sensitivity of the incremental equation of equilibrium is of the form:

$$\int_{\Omega'} \mathbf{B}_L^T \frac{d\Delta \mathbf{S}}{dh} d\Omega^t = \frac{\partial \Delta \mathbf{Q}}{\partial h}. \quad (7.4.8)$$

Taking into account the explicitly integrated constitutive equation the design derivative of the stress increment $\Delta \mathbf{S}$ is calculated. The elastic-plastic matrix $\mathbf{C}^{(e-p)}$ and the strains increment depend on the sensitivity to parameter h . Simultaneously, the elasto-plastic matrix is a function of the total stresses \mathbf{S} and internal variables vector $\boldsymbol{\gamma}$. The last two variables are known at time t , i.e. at the beginning of the step. Then, the stress increment takes the form:

$$\Delta \mathbf{S} = \mathbf{C}^{(e-p)}({}^t \mathbf{S}, {}^t \boldsymbol{\gamma}, h) \Delta \mathbf{e}(h). \quad (7.4.9)$$

Differentiating the Eq. (7.4.9) with respect to the parameter h the relation for the stress increment sensitivity takes the form:

$$\frac{d\Delta \mathbf{S}}{dh} = \left\{ \frac{\partial \mathbf{C}^{(e-p)}}{\partial^t \mathbf{S}} \frac{d^t \mathbf{S}}{dh} + \frac{\partial \mathbf{C}^{(e-p)}}{\partial^t \boldsymbol{\gamma}} \frac{d^t \boldsymbol{\gamma}}{dh} + \frac{\partial \mathbf{C}^{(e-p)}}{\partial h} \right\} \Delta \mathbf{e} + \mathbf{C}^{(e-p)}({}^t \mathbf{S}, {}^t \boldsymbol{\gamma}, h) \frac{d\Delta \mathbf{e}}{dh}. \quad (7.4.10)$$

Substituting the stress increment sensitivity in the (3.1.8) for the (3.1.10) and having in mind the written in finite element format the strains-displacements relation

$$\Delta \mathbf{e} = \mathbf{B}_L^T \Delta \mathbf{q}, \quad (7.4.11)$$

the expression for the displacement sensitivity increment is obtained:

$$\left(\int_{\Omega'} \mathbf{B}_L^T \mathbf{C}^{(e-p)} \mathbf{B}_L d\Omega' \right) \frac{d\Delta \mathbf{q}}{dh} = \frac{\partial \Delta \mathbf{Q}}{\partial h} - \int_{\Omega'} \mathbf{B}_L^T \frac{d\Delta \mathbf{S}}{dh} \Big|_{\Delta \mathbf{q}(h)=const} d\Omega'. \quad (7.4.12)$$

The first component of the above expression (7.4.12) (in the round brackets) stands for the tangent stiffness (elastic-plastic), and the last one is the design derivative of the nodal internal forces assuming that the displacement increment is independent of the parameter h . After solving the Eq. (7.4.12) for the displacement sensitivity increment, the sensitivity increments have to be accumulated in time, as well as the stress and the internal variables increments and their sensitivities.

Adjoint variable method (AVM)

An alternative way to calculate the sensitivity of the function (7.4.6) is the adjoint variable method (AVM). The objective of the method is to avoid the calculation of the displacement design derivative directly. The method may rationally be applied only to the problems of (linear or nonlinear) elasticity problems [27].

Additionally to the original structure an adjoint structure is defined. The adjoint structure has the same physical properties but is loaded by the adjoint load defined as the partial derivative of the function (7.4.6), with respect to the corresponding displacements. For the detailed description of the method, we refer to the literature [4, 26].

7.4.5. Constitutive model

An outline of the parameter sensitivity algorithm for a constitutive model having its application in the RC structures is presented herein. The Chan-Scordelis constitutive model for concrete [13, 19], and elastic-plastic with isotropic hardening constitutive model for steel are taken into account.

The main assumption of the model is the concept of uniaxial equivalent strains which allow to describe the material behavior in the principal directions [20]. Concerning the plane stress case, the possible states of the material are as follows: compression-compression, compression-tension and tension-tension. The relations for the compression-compression state are presented below. The uni-axial compression state describes by Saenz's equations [21]:

$$\sigma_i = \frac{E_o \varepsilon_{iu}}{1 + \left(\frac{E_o}{E_s} - 2 \right) \frac{\varepsilon_{iu}}{\varepsilon_{ic}} + \left(\frac{\varepsilon_{iu}}{\varepsilon_{ic}} \right)^2}, \quad (7.4.13)$$

$$E_i = \frac{E_o \left[1 - \left(\frac{\varepsilon_{iu}}{\varepsilon_{ic}} \right)^2 \right]}{\left[1 + \left(\frac{E_o}{E_s} - 2 \right) \frac{\varepsilon_{iu}}{\varepsilon_{ic}} + \left(\frac{\varepsilon_{iu}}{\varepsilon_{ic}} \right)^2 \right]^2}, \quad (7.4.14)$$

where E_o is the initial tangent modulus, E_s is the secant modulus, ε_{iu} are the equivalent uniaxial strains, ε_{ic} are the maximum equivalent uniaxial strains and σ_{ic} are the maximum equivalent uniaxial stresses in the plane stress state. The relations for σ_{ic} and for ε_{ic} are determined by the Kupfer-Gerstle curve [22]:

$$\begin{aligned} \sigma_{2c} &= \frac{1 + 3.65\beta}{(1 + \beta)^2} f_c, & \varepsilon_{ic} &= \varepsilon_c \left(3 \frac{\sigma_{2c}}{f_c} - 2 \right), \\ \sigma_{1c} &= \beta \sigma_{2c}, & \varepsilon_{2c} &= \varepsilon_c \left[-1.6 \left(\frac{\sigma_{1c}}{f_c} \right)^3 + 2.25 \left(\frac{\sigma_{1c}}{f_c} \right)^2 + 0.35 \left(\frac{\sigma_{1c}}{f_c} \right) \right], \end{aligned} \quad (7.4.15)$$

where β is the principal stresses ratio σ_1/σ_2 , f_c and ε_c are the maximum strength and maximum strains according to the uniaxial compression test. The Eqs. (7.4.13)–(7.4.15) are validated experimentally.

The right hand side of the sensitivity equation for the reinforced concrete in the plane stress state is as follows:

$$\frac{d\Delta\mathbf{F}}{dh} \Big|_{\Delta\mathbf{q}(h)=\text{const}} = \left(\int_{\Omega'} \mathbf{B}_L^T \frac{d\mathbf{C}(E_1, E_2, \nu)}{dh} d\Omega' \right) \Delta\mathbf{q} + \left(\int_{\Omega'} \mathbf{B}_L^T \frac{d\mathbf{C}^{(e-p)}}{dh} d\Omega' \right) \Delta\mathbf{q}. \quad (7.4.16)$$

The matrix \mathbf{C} is the constitutive matrix of the orthotropic material depends on the actual tangent moduli in the principal directions i . The matrix $\mathbf{C}^{(e-p)}$ is the elastic-plastic matrix for the reinforcing steel acting in the uniaxial direction.

Aiming at calculation of the parameter sensitivity of the increment of the internal forces vector, the sensitivities of the tangent uniaxial moduli that depend on the state of the material, *i.e.* total stresses and total strains and their derivatives have to be known. The stress increment in the direction i may be expressed as follows:

$$\Delta\sigma_i = E_i(\varepsilon_{iu}, \varepsilon_{ic}(\beta, h), \sigma_{ic}(\beta, h))\Delta\varepsilon_{iu}. \quad (7.4.17)$$

The parameter derivative of the stress increment takes the form:

$$\frac{d\Delta\sigma_i}{dh} = E_i \frac{d\Delta\varepsilon_{iu}}{dh} + \frac{dE_i}{dh} \Delta\varepsilon_{iu}. \quad (7.4.18)$$

The sensitivity of the tangent moduli is of the form:

$$\frac{dE_i}{dh} = \frac{\partial E_i}{\partial \varepsilon_{iu}} \frac{d\varepsilon_{iu}}{dh} + \frac{\partial E_i}{\partial \varepsilon_{ic}} \left(\frac{\partial \varepsilon_{ic}}{\partial \beta} + \frac{\partial \varepsilon_{ic}}{\partial h} \right) + \frac{\partial E_i}{\partial \sigma_{ic}} \left(\frac{\partial \sigma_{ic}}{\partial \beta} \frac{d\beta}{dh} + \frac{\partial \sigma_{ic}}{\partial h} \right) + \frac{\partial E_i}{\partial h}. \quad (7.4.19)$$

The partial derivatives in (7.4.19) can be obtained by differentiating the expressions of Kupfer-Gerstle curve (7.4.13)–(7.4.15). The ordinary derivatives are accumulated in time.

7.4.6. Applied finite element

In our application, the isoparametric layered shell element of Ahmad is used [14,15]. To have a consistent derivation, firstly the stiffness matrix of the element is presented, secondly, the contribution of the reinforcement to the element stiffness is shown and finally the sensitivities of the total stiffness matrix with respect to the parameters of the reinforcement properties, e.g., Young's moduli, thickness and the distance of the reinforcement layer from the midsurface of the shell are presented.

The geometrical relations in the shell element are defined in the four reference systems, Fig. 7.4.1. The systems are as follows:

- Global system – this is an arbitrary Cartesian system $\{X, Y, Z\}$, the final total stiffness is defined in that system.
- Local isoparametric system $\{\xi, \eta, \zeta\}$ – the curved midsurface of the element (in the ξ, η coordinates) and the positions of the layers (by the ζ coordinate) are defined.
- Local nodal coordinate systems – these systems are defined in each node by three orthogonal vectors $\{\mathbf{v}_1^i, \mathbf{v}_2^i, \mathbf{v}_3^i\}$ where i is the node number $i = 1..9$.
- Local coordinate system $\{x', y', z'\}$ – this is a Cartesian system in which the strains, stresses, generalized forces and the design derivatives are calculated. That system is defined at the integration stations

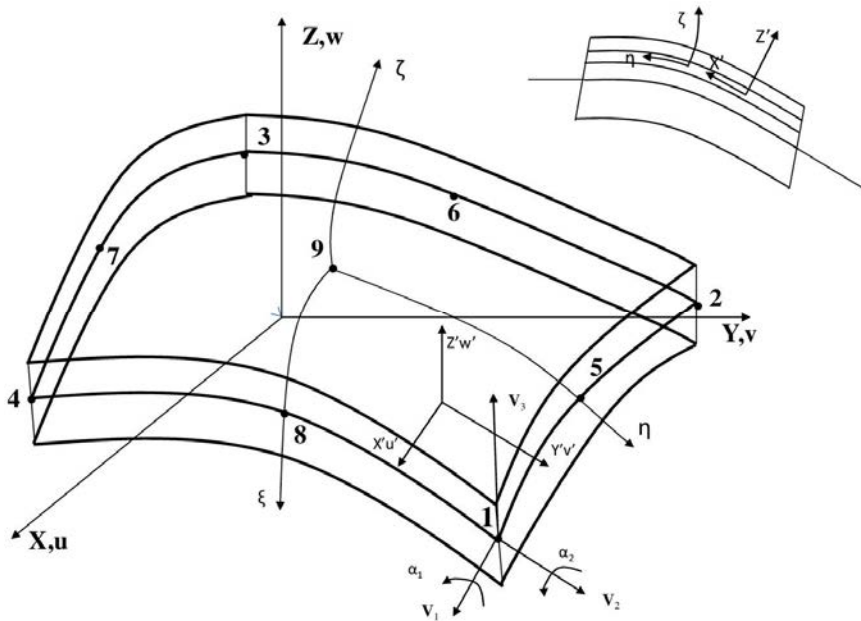


Figure 7.4.1. Layered shell element, coordinate systems.

The geometry of the element is defined by the relations:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \sum_{i=2}^n N_i \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \sum_{i=1}^n N_i \zeta \frac{t_i}{2} \mathbf{V}_3^i. \quad (7.4.20)$$

The displacement field in the element is defined by the following equation:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \sum_{i=2}^n N_i \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} + \sum_{i=1}^n N_i \zeta \frac{t_i}{2} \mathbf{\Phi}_i \begin{Bmatrix} \alpha_{1i} \\ \alpha_{2i} \end{Bmatrix}. \quad (7.4.21)$$

where N_i are the Lagrange shape functions and $\mathbf{\Phi}_i$ is a vector containing vectors tangent to the midsurface defined in the nodal coordinate system as follows:

$$\mathbf{\Phi}_i = [-\mathbf{V}_{2i}, \mathbf{V}_{1i}]. \quad (7.4.22)$$

At each node of the element three translational degrees of freedom u, v, w parallel to the axes of the global coordinate system X, Y, Z and two rotational degrees of freedom α_1, α_2 being the rotations about the vectors $\mathbf{V}_1, \mathbf{V}_2$ are defined. The geometrical relations developed on the local coordinate system are of the form:

$$\begin{bmatrix} e_{x'x'} \\ e_{y'y'} \\ e_{x'y'} \\ e_{x'z'} \\ e_{y'z'} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x'} & 0 & 0 \\ 0 & \frac{\partial}{\partial y'} & 0 \\ \frac{\partial}{\partial y'} & \frac{\partial}{\partial x'} & 0 \\ \frac{\partial}{\partial z'} & 0 & \frac{\partial}{\partial x'} \\ 0 & \frac{\partial}{\partial z'} & \frac{\partial}{\partial y'} \end{bmatrix} \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix}, \quad (7.4.23)$$

where

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \mathbf{\Theta}^T \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad \mathbf{\Theta} = [\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3]. \quad (7.4.24)$$

The above relation constitutes the dependence between the displacements in the global coordinate system X, Y, Z and the local one x', y', z' . Taking into consideration the displacement field (7.4.21) and the geometrical relations (7.4.23) and denoting by \mathbf{A} the following term:

$$\mathbf{\Theta}^T \mathbf{J}^{-1} = \mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix}, \quad (7.4.25)$$

where \mathbf{J} is the Jacobian of the transformation between the local coordinate system x', y', z' and the isoparametric one the geometrical relations take the form:

$$\mathbf{e}' = \mathbf{B}_i \mathbf{\Theta}^T \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} + \frac{t_i}{2} (\zeta \mathbf{B}_i + \mathbf{D}_i) \mathbf{\Theta}^T \mathbf{\Phi}_i \begin{bmatrix} \alpha_{1i} \\ \alpha_{2i} \end{bmatrix}. \quad (7.4.26)$$

Matrices \mathbf{B}_i and \mathbf{D}_i are of the form:

$$\mathbf{B}_i = \begin{bmatrix} B_1 & 0 & 0 \\ 0 & B_2 & 0 \\ B_2 & B_1 & 0 \\ 0 & 0 & B_1 \\ 0 & 0 & B_2 \end{bmatrix}, \quad \mathbf{D}_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ D_1 & 0 & 0 \\ 0 & D_1 & 0 \end{bmatrix}, \quad (7.4.27)$$

where

$$\begin{aligned} B_1 &= A_{11}N_{i,\xi} + A_{12}N_{i,\eta}, \\ B_2 &= A_{21}N_{i,\xi} + A_{22}N_{i,\eta}, \\ D_1 &= A_{33}N_i. \end{aligned} \quad (7.4.28)$$

The stiffness matrix is obtained using numerical integration. The element is layered and is integrated using Simpson's rule across the thickness and 4 or 9 points Gauss rule over the surface. The ready for programming stiffness matrix is of the form:

$$\begin{aligned} \mathbf{K}_E^{ij} &= \sum_{m=1}^N \sum_{n=1}^{nc+1} w_m w_n^c \begin{bmatrix} \mathbf{\Theta}^T & 0 \\ 0 & \frac{t_i}{2} \mathbf{\Theta}^T \mathbf{\Phi}_j \end{bmatrix}_m^T \left[\mathbf{B}_i | \zeta^c \mathbf{B}_i + \mathbf{D}_i \right]_m^T \\ &\quad \cdot \mathbf{C}_{mn}^c \left[\mathbf{B}_j | \zeta^c \mathbf{B}_j + \mathbf{D}_j \right]_m \begin{bmatrix} \mathbf{\Theta}^T & 0 \\ 0 & \frac{t_j}{2} \mathbf{\Theta}^T \mathbf{\Phi}_j \end{bmatrix} \det \mathbf{J}_m. \end{aligned} \quad (7.4.29)$$

The \mathbf{C}_{mn}^c is the constitutive matrix for the orthotropic material defined at the integration stations, the summation over m is done over integration points. N depends on the integration rule is 4 or 9. Summation over n is done across the thickness, nc is the number of the concrete layers in the element, w_m is the Gauss weight, ζ is the normalized coordinate of the layer n measured in the direction ζ , w_n^c is the weighting coefficient for the Simpson rule, t_m is the thickness of the shell at the point m .

Reinforcement

The 'smeared' model of the reinforcement is used. The equivalent thickness $t_s = A_s d_s$, where A_s is the cross-sectional area of the reinforcement bar and d is the distance between the bars, is assumed. The uniaxial stress state is assumed in the reinforcing bars. The reinforcement is placed in the two orthogonal directions. Then, the constitutive matrix for the reinforcement assuming its existence in both directions is of the form:

$$\mathbf{C}'_s = \begin{bmatrix} E_s & 0 & 0 & 0 & 0 \\ & E_s & 0 & 0 & 0 \\ & & 0 & 0 & 0 \\ & & & 0 & 0 \\ & & & & 0 \end{bmatrix}. \quad (7.4.30)$$

If the reinforcement exists only in one chosen direction the nonzero elements of the above matrix (7.4.30) are the relevant or $C'_{s(11)}=E_s$ or $C'_{s(22)}=E_s$ terms (E_s is the Young's modulus for the reinforcement). The total stiffness is the sum of the stiffness of the concrete and the equivalent reinforcement layers:

$$\mathbf{K}_E = \mathbf{K}_c + \mathbf{K}_s. \quad (7.4.31)$$

The stiffness matrix of the reinforcement is similar to the stiffness matrix (7.4.29) presented above. It is obtained introducing instead of the constitutive matrix for the orthotropic material the relevant constitutive matrices for the equivalent reinforcement layers (7.4.30). The stiffness matrix for the reinforcement layers is of the form:

$$\mathbf{K}_s^{ij} = \sum_{m=1}^N \sum_{n=1}^{ns} w_m w_n^s \begin{bmatrix} \mathbf{\Theta}^T & 0 \\ 0 & \frac{t_i}{2} \mathbf{\Theta}^T \mathbf{\Phi}_j \end{bmatrix}_m^T \left[\mathbf{B}_i | \zeta^s \mathbf{B}_i + \mathbf{D}_i \right]_m^T \cdot \mathbf{C}_{mn}^c \left[\mathbf{B}_j | \zeta^s \mathbf{B}_j + \mathbf{D}_j \right]_m \begin{bmatrix} \mathbf{\Theta}^T & 0 \\ 0 & \frac{t_j}{2} \mathbf{\Theta}^T \mathbf{\Phi}_j \end{bmatrix} \det \mathbf{J}_m. \quad (7.4.32)$$

The summation over m index denotes the sum over Gauss points on the surface of the element. The ns is the number of the reinforcement layers, ζ_n^s is the non-dimensional coordinate that fixes the position of the reinforcement layer with respect to the midsurface of the shell element (direction ζ). The \mathbf{C}_{mn}^s is the constitutive matrix of the reinforcement layer at the integration point.

Sensitivities of the stiffness matrix to the reinforcement parameters

The formulae concerning the sensitivities of the total stiffness matrix to the selected parameters of the reinforcement are presented herein. The parameters are Young's modulus, thickness of the reinforcement layer and the distance of the reinforcement layer from the midsurface of the shell. To calculate the sensitivity of the total stiffness at the chosen layer, the components of the sensitivities at the reinforcement layer should be calculated and summed up. In the case of the Young's modulus, the sensitivity takes the form:

$$\frac{\partial \mathbf{K}_s^{ij}}{\partial E} = \sum_{m=1}^N w_m w_{il}^s \begin{bmatrix} \mathbf{\Theta}^T & 0 \\ 0 & \frac{t_i}{2} \mathbf{\Theta}^T \mathbf{\Phi}_j \end{bmatrix}_m^T \left[\mathbf{B}_i | \zeta^s \mathbf{B}_i + \mathbf{D}_i \right]_m^T \cdot \frac{\partial \mathbf{C}_{m(il)}^s}{\partial E} \left[\mathbf{B}_j | \zeta^s \mathbf{B}_j + \mathbf{D}_j \right]_m \begin{bmatrix} \mathbf{\Theta}^T & 0 \\ 0 & \frac{t_j}{2} \mathbf{\Theta}^T \mathbf{\Phi}_j \end{bmatrix} \det \mathbf{J}_m. \quad (7.4.33)$$

The sensitivity of the constitutive matrix (7.4.32) to the Young's modulus in the case of the existence of reinforcement in both orthogonal directions is of the form:

$$\frac{\partial \mathbf{C}_{m(il)}^s}{\partial E} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ & 1 & 0 & 0 & 0 \\ & & 0 & 0 & 0 \\ & & & 0 & 0 \\ & & & & 0 \end{bmatrix}. \quad (7.4.34)$$

If the reinforcement exists only in one selected direction, the relevant elements of the matrix given above are zeroed (element 11 or element 22). The formula for the sensitivity of the total stiffness matrix to the thickness of the reinforcement layer is obtained by differentiating of the formula (7.4.29) with respect to the thickness of the layer at each integration station in the layer. The component of the stiffness matrix connected with the reinforcement layer in the element is linearly dependent on the weighting factor of the rectangular integration rule. In consequence, the sensitivity of the total stiffness to the thickness of a particular layer takes the form:

$$\frac{\partial \mathbf{K}_s^{ij}}{\partial t_s} = \sum_{m=1}^N w_m \frac{2}{t_m} \begin{bmatrix} \mathbf{\Theta}^T & 0 \\ 0 & \frac{t_i}{2} \mathbf{\Theta}^T \mathbf{\Phi}_j \end{bmatrix}_m^T \left[\mathbf{B}_i | \zeta^s \mathbf{B}_i + \mathbf{D}_i \right]_m^T \cdot \mathbf{C}_{m(il)}^s \left[\mathbf{B}_j | \zeta^s \mathbf{B}_j + \mathbf{D}_j \right]_m \begin{bmatrix} \mathbf{\Theta}^T & 0 \\ 0 & \frac{t_j}{2} \mathbf{\Theta}^T \mathbf{\Phi}_j \end{bmatrix} \det \mathbf{J}_m. \quad (7.4.35)$$

Finally, in the case of the distance of the reinforcement layer from the midsurface of the shell the sensitivity derivative of the stiffness matrix with to this parameter is of the form:

$$\frac{\partial \mathbf{K}_s^{ij}}{\partial \zeta^{il}} = \sum_{m=1}^N w_m w_{il}^s \left[\begin{array}{cc} \boldsymbol{\Theta}^T & 0 \\ 0 & \frac{t_i}{2} \boldsymbol{\Theta}^T \boldsymbol{\Phi}_j \end{array} \right]_m^T$$

$$\cdot \left[\begin{array}{cccccc} 0 & 0 & 0 & B_1^{(i)} B_1^{(j)} E_s^{(il)} & 0 & 0 \\ 0 & B_2^{(i)} B_2^{(j)} E_s^{(il)} & 0 & 0 & B_2^{(i)} B_2^{(j)} E_s^{(il)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ B_1^{(i)} B_1^{(j)} E_s^{(il)} & 0 & 2\zeta^{il} B_1^{(i)} B_2^{(j)} E_s^{(il)} & 0 & 0 & 0 \\ 0 & B_2^{(i)} B_2^{(j)} E_s^{(il)} & 0 & 0 & 2\zeta B_2^{(i)} B_2^{(j)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right]_m$$

$$\cdot \left[\begin{array}{cc} \boldsymbol{\Theta}^T & 0 \\ 0 & \frac{t_j}{2} \boldsymbol{\Theta}^T \boldsymbol{\Phi}_j \end{array} \right] \det \mathbf{J}_m. \quad (7.4.36)$$

The expressions for the stiffness sensitivity are substituted into Eq. 7.4.12. Having the displacement sensitivity field we may easily obtain the stress sensitivities, considering the formula for strains (7.4.26) and taking into account the constitutive relations for the reinforcement (7.4.34), it reads:

$$\frac{d\boldsymbol{\sigma}}{dh} = \frac{d\mathbf{C}_{m^{(il)}}^s}{dh} \left[\begin{array}{cc} \boldsymbol{\Theta}^T & 0 \\ 0 & \frac{t_i}{2} \boldsymbol{\Theta}^T \boldsymbol{\Phi}_i \end{array} \right] \mathbf{q} + \mathbf{C}_{m^{(il)}}^s \left[\begin{array}{cc} \boldsymbol{\Theta}^T & 0 \\ 0 & \frac{t_j}{2} \boldsymbol{\Theta}^T \boldsymbol{\Phi}_j \end{array} \right] \frac{d\mathbf{q}}{dh}. \quad (7.4.37)$$

The h is the reinforcement parameter (Young's modulus). The stress sensitivities to the parameters of the concrete matrix may be obtained setting the constitutive matrix for orthotropic material in the formula (7.4.37) above.

7.4.7. Computer program

The computer program consists of two main modules, namely, the reliability module solving the problem defined by the (7.4.2) and the module solving the static equilibrium problem formulated by (7.4.8) with implemented parameter sensitivity analysis. The parameter sensitivity is necessary for the calculation of the sensitivities to the standardized variables given by the (7.4.4). The general parameter sensitivity formulation is expressed by (7.4.5), (7.4.6). The reliability module is the program COMREL-TI, [9]. The nonlinear static equilibrium problem and the parameter sensitivity problem (7.4.8) are solved by the significantly extended version of the program NASHL [13]. The adjoint variable method is used for the linear problems and the direct differentiation method for the nonlinear ones.

7.4.8. McNeice plate

The dimensions in the plane of the square plate McNeice (non-dimensional units) are 36x36, and its thickness is 1.75. The equivalent thickness of the orthotropic reinforcement is 0.11 (the same in both directions). The distance of the reinforcement from the lower surface of the plate is 0.44. The initial Young modulus for concrete is 4.15E+6. The ultimate strength in compression is 5.5E+3. Young's modulus for steel is 2.9E+7 and yield limit is 6.0E+4. The plate is loaded in its midspan with the point force 5000.0 and is simple supported at the corners. The system is discretized with the described above 36 isoparametric 9-node, layered, plate-shell Ahmad-type elements [12], Fig. 7.4.1.

The stochastic model consists of four stochastic parameters connected with the reinforcement and load multiplier. The stochastic variables are defined as thicknesses of the reinforcement in four quads of the plate. The distributions of the thicknesses are standard with the standard deviations 5%, 10%, 10% and 15% counted starting from the left lower quad counterclockwise. The distribution if the load multiplier is log-normal with the standard deviation 20%. The design constraint is set on the vertical displacement of the midplate, and the displacement should not exceed 0.16.

The displacement field is presented in Fig.7.4.2a, and the displacement sensitivity field is also given in Fig. 7.4.2b. The displacement sensitivity field is given for that case when the design parameter is the equivalent thickness of the steel layer in the right, upper quad of the plate. The fields presented in Fig. 7.4.2 are valid for the load level 73% of the total load and the maximum deterministic displacement 0.15.

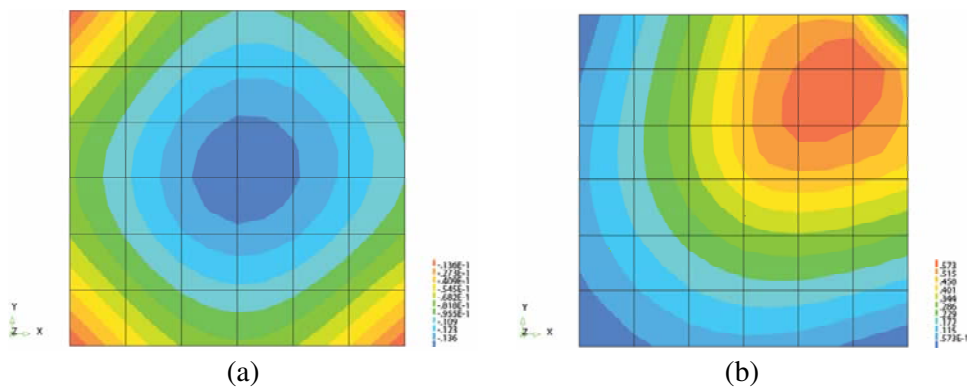


Figure 7.4.2. McNeice plate: displacement field (a) and displacement sensitivity field (b), parameter – thickness of the reinforcement layer in the right, upper quad of the plate.

For that stochastic system, the reliability index β is 4.283, and the corresponding probability of failure is 9.2376E-6.

7.4.9. Nuclear containment shell

Description of the structure

The next analysed structure is a nuclear containment vessel [14, 15, 23]. The scheme of the structure is presented in Fig. 7.4.3. The structure consists of a cylinder (radius 20 m) and dome, height of the structure is 64 m. The vessel is discretized with 640 shell elements. The mesh is presented in Fig. 7.4.4a. The structure is divided into seven zones along its height depending on the reinforcement distribution in each zone. The scheme of the division is also given in Fig. 7.4.4b.

The circumference of the structure is divided into 20 elements. The divisions along the height are different in each zone, namely, the number of elements in the zone I is 40, in zone II – 40, in zone III – 80, in zone IV – 40, in zone V – 1180, in zone VI – 120 and zone VII – 40. The number of nodes is 2600 and number of d.o.f. is 12800.

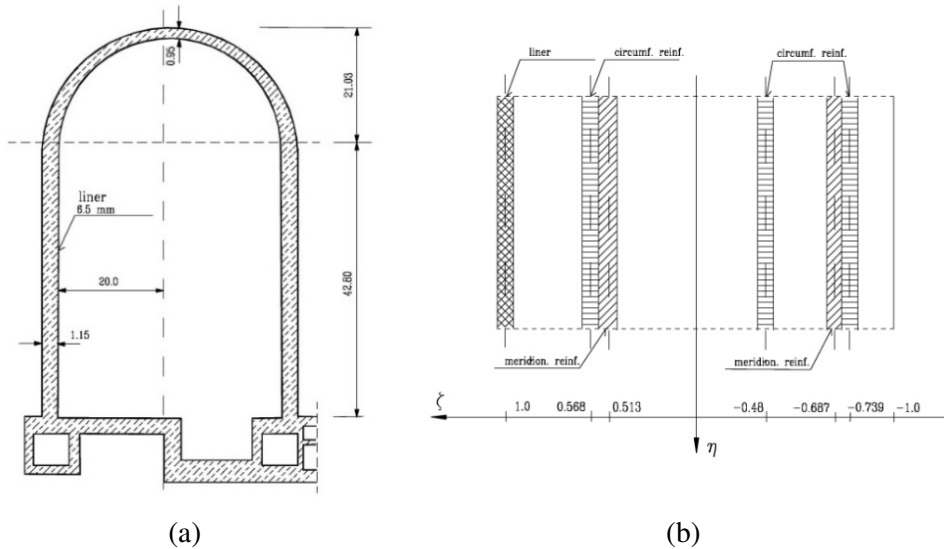


Figure 7.4.3. Containment vessel, vertical cross-section (a) and a typical cross-section of the wall (b).

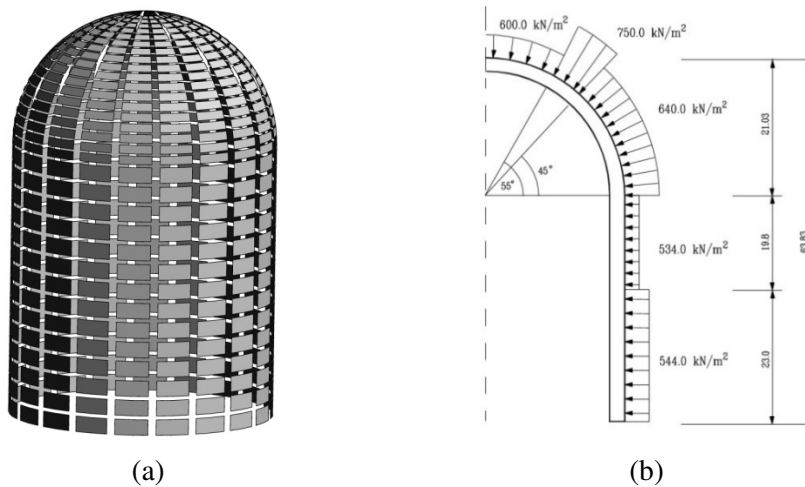


Figure 7.4.4. Finite element discretization, shrink plot (a), reinforcement and external pressure zones (b).

The structure is prestressed with a system of tendons, and the action of the active reinforcements is replaced by the action of the external pressure. The external pressure distribution is approximately in compliance with the zoning division. Starting from the base the first 160 elements is loaded with the pressure 544 kN/m², the next 60 elements are loaded with the pressure 534 kN/m² and on the next 260 elements acts the pressure 640 kN/m². The elements in the upper part of the dome are loaded with the pressure 600 kN/m² (the first 120 from the top of the dome), and the next 40 elements placed close to the upper edge of the cylinder is loaded with the pressure 720 kN/m². The data concerning the reinforcement distribution is given in Table 7.4.1.

Young's modulus of the reinforcement steel of the liner and the passive reinforcement is 2.1E+8 kN/m² and the prestressed reinforcement is 1.025E+8 kN/m².

The yield limits of the steel used for liner, passive and prestressed reinforcement are 1.68E+5 kN/m², 4.2E+5 kN/m² and 3.2E+5 kN/m², respectively. The hardening modulus for all types of steel is 3.0E+7 kN/m². Initial Young's modulus for concrete is 3.0E+7 kN/m², yield limit in compression is 3.2E+4 kN/m² strength in tension is 3.2E+3 kN/m².

Table. 7.4.1. Reinforcement data

Zone	Dist. from midsurface	Equivalent thickness	Description
I	1.000	0.650E-02	liner
	-0.739	0.761E-02	circ. ext.
	0.513	0.264E-02	circ. internal
	-0.687	0.940E-02	meridional ext.
	0.565	0.974E-02	meridional int.
	-0.480	0.900E-02	prestressed
II	1.000	0.650E-02	liner
	-0.739	0.761E-02	circ. ext.
	0.513	0.264E-02	circ. internal
	-0.687	0.530E-02	meridional ext.
	0.565	0.309E-02	meridional int.
	-0.480	0.900E-02	prestressed
III	1.000	0.650E-02	liner
	-0.739	0.761E-02	circ. ext.
	0.513	0.264E-02	circ. internal
	-0.687	0.530E-02	meridional ext.
	0.565	0.193E-02	meridional int.
	-0.480	0.900E-02	prestressed
IV	1.000	0.650E-02	liner
	-0.739	0.761E-02	circ. ext.
	0.513	0.264E-02	circ. internal
	-0.687	0.414E-02	meridional ext.
	0.565	0.193E-02	meridional int.
	-0.480	0.900E-02	prestressed
V	1.000	0.650E-02	liner
	-0.725	0.471E-02	circ. ext.
	0.537	0.275E-02	circ. internal
	-0.725	0.448E-02	meridional ext.
	0.537	0.217E-02	meridional int.
	-0.480	0.900E-02	prestressed
VI	1.000	0.650E-02	liner
	-0.725	0.328E-02	circ. ext.
	0.537	0.204E-02	circ. internal
	-0.725	0.328E-02	meridional ext.
	0.537	0.204E-02	meridional int.
	-0.480	0.900E-02	prestressed
VII	1.000	0.650E-02	liner
	-0.725	0.328E-02	circ. ext.
	0.537	0.275E-02	circ. internal
	-0.725	0.328E-02	meridional ext.
	0.537	0.217E-02	meridional int.
	-0.480	0.900E-02	prestressed

Linear problem

The adjoint variable method is very well suitable to solve linear problems with a high number of design variables [26]. In consequence, the method is very efficient when attempting to solve this type of reliability problems.

An example of such a stochastic system is created. The stochastic parameters are the thicknesses of the reinforcement layers in particular elements and load multiplier. Normal distributions are assumed for the steel layers with the standard deviations as follows: 15%, 15%, 1.5%, 15%, 15%, 16%, counting from the liner to the most external layer. For the load multiplier the log-normal distribution is assumed, and the standard deviation is 20%. The whole stochastic model consists of 3841 variables. The structure is loaded with the external equivalent pressure, and the reliability index is evaluated for the internal pressure load multiplier 6.5.

The failure function is described by the excessive horizontal displacement of a point (651) in the midspan of the cylinder (7.4.38) with allowable value $0.15E-2$ m.

$$g(x) = q^* - q_{651}^h. \quad (7.4.38)$$

The deterministic solution is presented in Fig. 7.4.5. The shape of the structure and the parameter sensitivity field of the chosen horizontal displacement are shown. The parameters variables are the thicknesses of the liner in particular elements. The deterministic displacement is $0.4824E-3$ m and the highest absolute sensitivity gradients are in the neighbourhood of the constraint and on the sides (left and right) of the cylinder. The estimated reliability index is 6.08 and corresponds with the probability of violation of the allowable displacements $6.169E-10$. The beta index sensitivities with respect to mean values and standard deviations are presented in Fig. 7.4.6.

The distribution of the beta index sensitivities (Fig. 7.4.6) is qualitatively similar to the design sensitivities of the chosen displacement calculated during the deterministic solution. The distribution of the beta index derivatives with respect to the standard deviation is qualitatively different from the parameter sensitivities, the lowest beta index sensitivities correspond to the highest parameter sensitivities in the deterministic solution. A more extended analysis of the vessel in the linear range is given in [23].

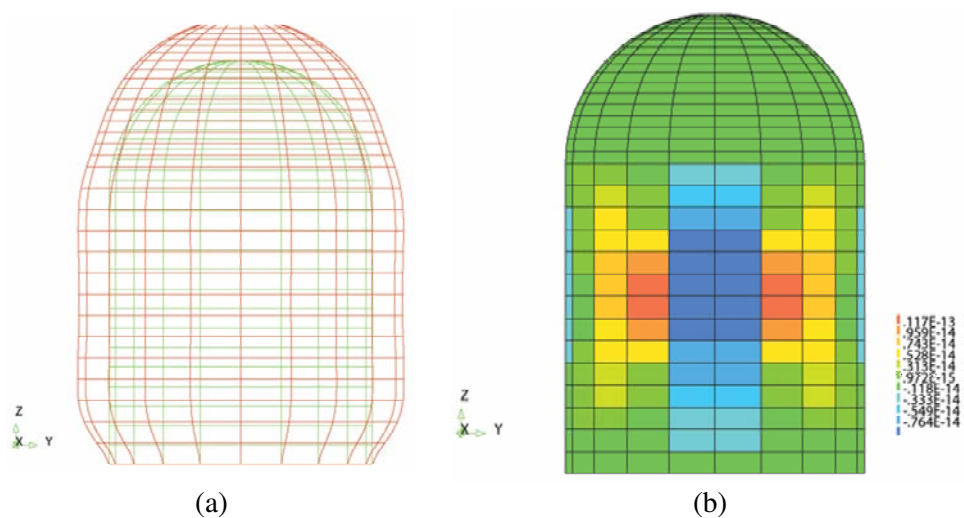


Figure 7.4.5. Displacement sensitivity with respect to thicknesses of the liner in particular elements, deterministic solution.

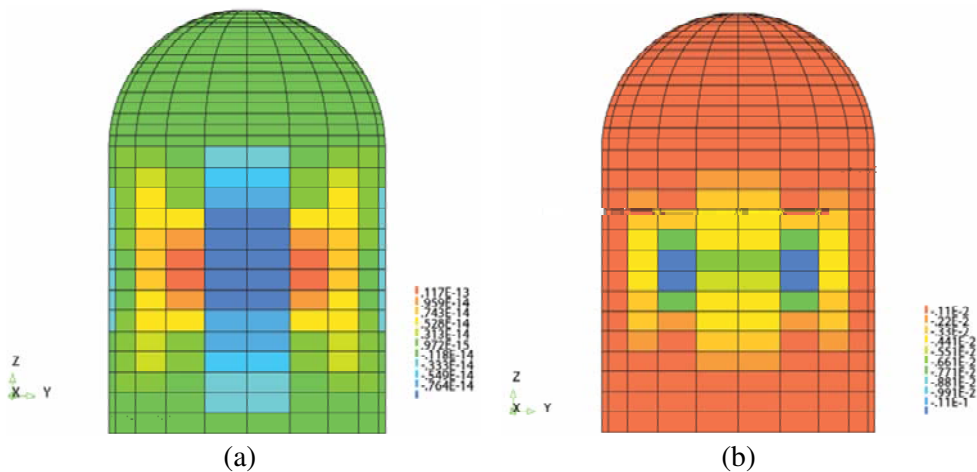


Figure 7.4.6. Reliability index sensitivity w.r.t. mean values (a) and standard deviations (b) of the thicknesses of the liner in particular elements.

Nonlinear problem, DDM

As previously, the goal of the analysis is the evaluation of the reliability of the system taking into account displacement failure function. The structure is investigated in the nonlinear range. The behaviour of concrete is described by the constitutive model presented above and steel is modelled using elastic-plastic model with isotropic

hardening. Aspects of the equilibrium analysis, design sensitivity and crude Monte-Carlo method have been described in Refs [14–17].

The sensitivity analysis of the vessel is carried out for a range of load starting from the prestressing phase (external pressure) passing to the increase of uniform internal pressure up to failure. The parameter is the thickness of the internal circumferential reinforcement layer in a chosen element (141).

The results of the sensitivity analysis are presented in Fig. 7.4.7 and 7.4.8. The shape of the structure in the prestressing phase is presented in Fig. 7.4.7 a, and the corresponding design sensitivity field is given in Fig. 7.4.7b. The shape of the structure in the failure phase is given in Fig. 7.4.8a, and the displacement sensitivity field is shown in Fig. 7.4.8b. The corresponding internal pressure is 10.64 kN/m^2 . The horizontal displacement of a point in the midspan of the cylinder is 0.174 m . A qualitative difference is manifested when comparing the sensitivity fields for both phases. The horizontal displacement of a point in the midspan of the cylinder is 0.174 m . A qualitative difference is manifested when comparing the sensitivity fields for both phases. During the prestressed phase, the behaviour of the structure is almost linear the highest values of the sensitivity gradients are concentrated mostly close to the investigated element. In the failure phase, when the nonlinear behaviour of the material is very significant, the high values of the gradients are distributed over the whole structure.

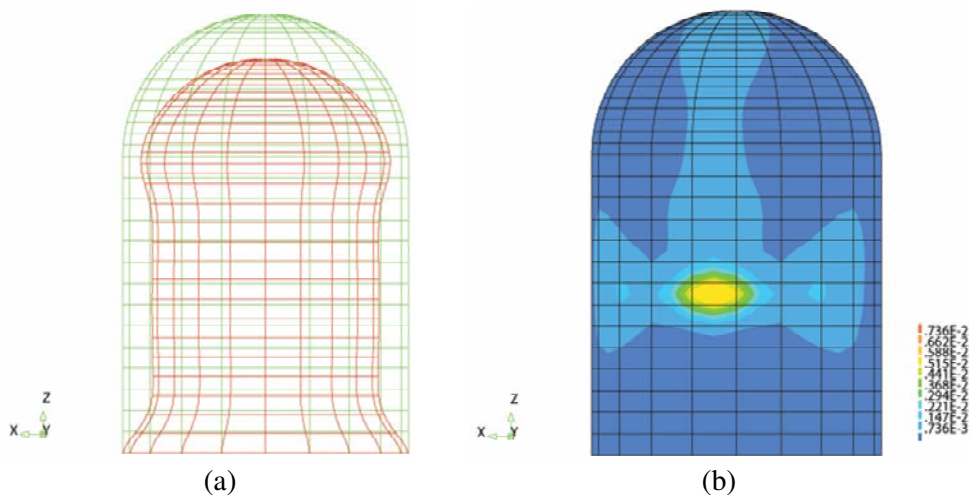


Figure 7.4.7. Shape of the structure, prestressing phase (a), displacement sensitivity field (b), design parameter – thickness of reinforcement layer, internal, circumferential, element 141.

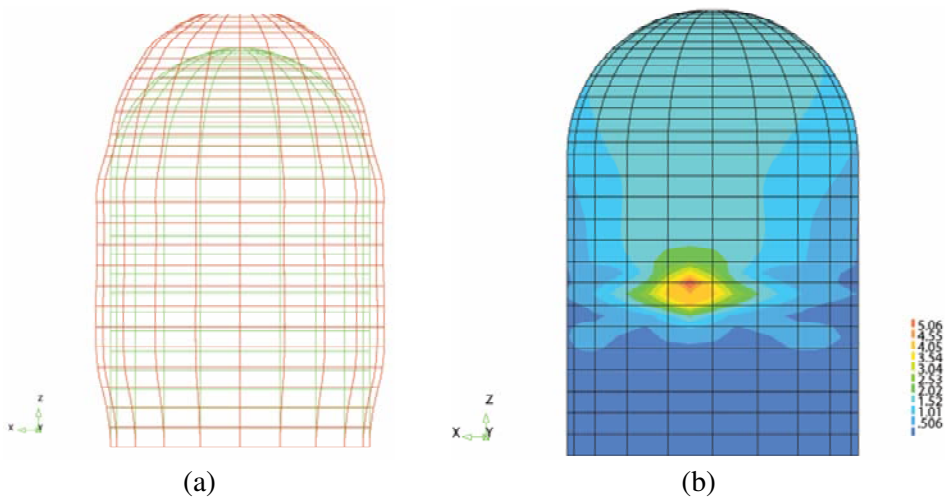


Figure 7.4.8. Increase of the internal pressure, shape of the structure (a), displacement sensitivity field (b), design parameter – thickness of reinforcement layer, internal, circumferential, element 141.

In the next step, the reliability analysis of the system is performed. The stochastic parameters are the distances of whole reinforcement layers from the midsurface. The stochastic system consists of 6 parameters connected with the reinforcement and load multiplier. Normal distributions are assumed for the distances from the midsurface of the shell with the following standard deviations: 0.8%, 0.4%, 0.5%, 0.5%, 0.1%, 0.5%. The log-normal distribution with the standard deviation 20% is assumed for the load multiplier.

The displacement failure function is considered. The design constraint is set on the horizontal displacement of a point 651 in the midspan of the cylinder (7.4.38). The horizontal displacement should not exceed 0.035 m. The shape of the structure is given in Fig. 7.4.9a, and the design sensitivity field from the deterministic solution is presented in Fig. 7.4.9b. The reliability analysis is carried out for load level 415 kPa. The deterministic displacement for this load level is 0.186E-3 m. The obtained reliability index is 6.561.

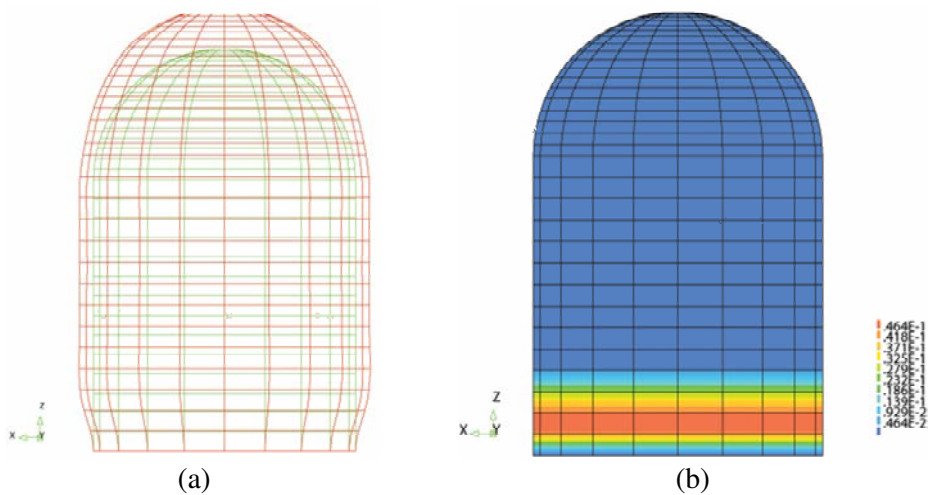


Figure 7.4.9. Increase of the internal pressure phase, shape of the structure (a) and the displacement sensitivity field (b), design parameter – distance of the liner from the midsurface.

7.4.10. Final remarks

A short description of an algorithm concerning the reliability index evaluation is described. The sensitivity coefficients are calculated employing the adjoint variable and the direct differentiation method along with the analytical approach. The effectiveness of the algorithm is proved by presenting the numerical examples. The computational effort is still very high; however it may be decreased by further improvement of the efficiency of the equilibrium and the sensitivity analyses using the vectorized and parallel solvers.

References

- [1.] P. Marek, M. Gustar, and T. Anagnos. *Simulation-based reliability assesment for structural engineers (new directions in civil engineering)*. CRC Press, 1995.
- [2.] M. Kleiber and T. D. Hien. *The stochastic finite element method: basic perturbation technique and computer implementation*. Wiley & Sons, 1992.
- [3.] E. J. Haug, K. K. Choi, and V. Komkov. *Design Sensitivity Analysis of Structural Systems*. Academic Press, New York, 1986.
- [4.] M. Kleiber, H. Antunez, T. D. Hien, and P. Kowalczyk. *Parameter Sensitivity in Nonlinear Mechanics*. Wiley & Sons, 1997.
- [5.] H. O. Madsen, S. Krenk, and N. C. Lind. *Methods of Structural Safety*. Prentice-Hall, 1986.

- [6.] J. L. T. Santos, A. Siemaszko, S. Gollwitzer, and R. Rackwitz. Continuum sensitivity method for reliability-based structural design and optimization. *Mech. Struct. & Mach.*, 23:497–520, 1995.
- [7.] A. Siemaszko and J. L. T. Santos, Reliability-based structural optimization. *Proc. Struct. Opt.* 93 World Congress, Rio de Janeiro, I:473–480, 1993.
- [8.] A. M. Hasofer and N. C. Lind. Exact and invariant second moment code format. *Journal of Engineering Mechanical Division*, ASCE, 100 (1):111–121, 1974.
- [9.] COMREL–TI, User's Manual, Reliability Consulting Programs GmbH, Barer Str. 48, Munich, GE., 1992.
- [10.] M. Hohenblicher and R. Rackwitz. First-order concepts in system reliability. *Structural Safety*, 1(3):177–188, 1982–1983.
- [11.] R. Rackwitz and B. Fiessler. Structural reliability under combined load sequences. *Computers & Structures*, 9 (5):489–494, 1978.
- [12.] S. Ahmad, B. M. Irons, and O. C. Zienkiewicz. Analysis of thick and thin shell structures by curved finite elements. *International Journal of Numerical Methods in Engineering*, 2(3):419–451, 1970.
- [13.] E. Y. Chan. *Nonlinear geometric, material and time dependent analysis of reinforced concrete shells with edge beams*. Technical report, University of California, Berkeley, 1982.
- [14.] E. Postek. Numerical parameter sensitivity of large nonlinear structural systems. Ph.D. dissertation, IFTR, Warsaw, PL, (in Polish), 1996.
- [15.] E. Postek and M. Kleiber. Parameter sensitivity of RC shell structures. Prof. Z. Kaczowski Jubilee Volume, Warsaw University of Technology, Warsaw, pp. 345–358, (in Polish), 1996.
- [16.] A. H. Barbat, M. Cervera, C. Cirauqui, A. Hanganu, and E. Onate. Evaluacion de la presion de fallo del edificio de contencion de una central tipo PWR--W tres lazos. Parte II: Simulacion numerica. *Revista Internacional de Met. Num. para Calculo y Diseno en Ingenieria*, 11 (3):451–475, 1995.
- [17.] E. Postek, A. Hanganu, E. Onate, and A. H. Barbat. Failure analysis of nuclear reactor containment shells. *30th Polish Conference on Solid Mechanics*, Conference Abstracts, Zakopane, September 1994.
- [18.] O. Ditlevsen and H. O. Madsen. *Structural reliability methods*. Wiley & Sons, 1996.
- [19.] C. S. Lin and A. C. Scordelis. Nonlinear analysis of reinforced concrete shells of general form. *J. Struc. Div.*, ASCE, 101 (3):523–538, 1975.
- [20.] D. Darwin and D. A. Peckhold. Nonlinear biaxial stress-strain law for concrete. *J. Eng. Mech. Div.*, ASCE, EM2, 1997.
- [21.] L. P. Saenz. Discussion of 'Equation for the stress strain curve of concrete' by Desayi P., *ACI Journal*, 61:9, 1964.
- [22.] H. B. Kupfer and K.H. Gerstle. Behaviour of concrete under biaxial stresses. *J. Eng. Mech. Div.*, ASCE, EM4, 1979.

- [23.]E. Postek, A. Siemaszko, and M. Kleiber. Reliability study of a containment shell. *Int. Jou. of Theoretical Mech.*
- [24.]K. J. Bathe. *Finite element procedures in engineering analysis*. Prentice–Hall Inc., New Jersey, 1995.
- [25.]M. Kleiber. Shape and non-shape structural sensitivity analysis form problems with any material and kinematic nonlinearity. *Comput. Meth. in Appl. Mech. Eng.*, 108:73–97, 1993.
- [26.]T. D. Hien and M. Kleiber. Computational aspects in structural design sensitivity analysis for statics and dynamics. *Comput. Struct.*, 33 (4):939–950, 1989.
- [27.]M. Kleiber, T. D. Hien, and E. Postek. Incremental finite element sensitivity analysis for non-linear mechanics applications. *International Journal of Numerical Methods in Engineering*, 32:1247–1265, 1991.
- [28.]A. D. Belegundu and J. S. Arora. A sensitivity interpretation of adjoint variables in optimal design. *Comp. Meth. Appl. Mech. Eng.*, 89:48–81, 1985.

7.5. Development of a concept of an agent-stress model of a tissue

Eligiusz Postek

7.5.1. Problem statement

We present a concept of coupling between agent modelling and stress analysis valid for a tissue. This is the development of the model presented in [1]. The growing epithelial tissue while treating each cell as an agent is shown in [2]. We use a tensegrity model for a single cell. This model serves for stress evaluation in growing tissue. We propose to couple the mechanical modelling with the agent one. This is done using FLAME (Flexible Large-scale Agent Modelling Environment). The rules implemented into the agent model, can be identified in the following categories; Signal rules (used to communicate cell location and internal state), Cell Cycle rules, Cell Division rules, Differentiation rules, Migration rules and Location Resolution rules. In previous work [3] on epithelial tissue, the keratinocyte colony formation model contains four types of cells, namely, Stem cells, Transit Amplifying cells, Committed cells and Corneocytes. The tissue evolution is steered by this model and the stress evolution is followed employing tensegrity model of the tissue.

The mechanical environment is important to the cells' behavior. It changes the cell performance. We stress the following statement: "Change the mechanical stresses experienced by cancer cells and they may start to behave more like healthy ones" [4]. However, the observation of stress development in a single cell and in particular, in

a growing tissue is still rare. The same concerns the numerical simulations. The growing epithelial tissue while treating each cell as an agent is modelled in [3]. We are showing a concept of coupling between the agent modelling which catches up several basic rules governing the tissue evolution and a mechanical model of a tissue. We believe that this idea should draw attention of the researchers to unconventional ways of coupling.

7.5.2. General algorithm.

The algorithm is based on the well-known idea of staggered solution [5]. The scheme is given in Fig. 7.5.1. In this case, the T-module is the agent module and the M-module is the mechanical one. The solution from the agent module is transferred to the mechanical module and the feedback from the mechanical module is transferred to the agent module for the next time step.

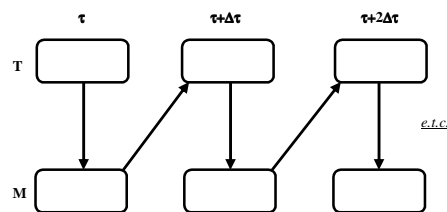


Figure 7.5.1. Staggered scheme.

This kind of coupling was applied to model thermomechanical and thermofluid phenomena [6, 7].

7.5.3. Characterization of the agent model

A brief characterization of the model is shown in this section. It follows [3] and the report [8]. Each cell is treated as an agent. The cells communicate by sending and receiving information while exchanging data on their type and positions. The cells act accordingly to a cell cycle. They divide, differentiate and migrate. Their functions depend on the calcium concentration in the environment. The simple division of stem cells is shown in Fig. 7.5.2. It can be seen looking at the most left three clusters of the cells. While the cluster becomes big enough the stem cells can differentiate into Transit Amplifying cells. An example of such situation is seen in Fig. 7.5.2 again in the most right cluster. The TA cell is indicated as a ring.

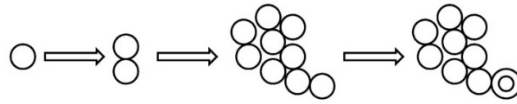


Figure 7.5.2. Stem cells differentiate into Transit Amplifying (TA) cells.

When the cluster of stem cells becomes surrounded by TA cells, the TA cells on the edge differentiate into Committed cells, Fig. 7.5.3a. They are shown as black rings. Additionally, the stem cells staying long time in contact can differentiate into Committed cells as well, Fig. 7.5.3b. The TA cells can migrate around the colony depending on the calcium levels. The stem cells stay statically. The Corneocytes are the dead TA and Committed cells.



Figure 7.5.3. Situations while Committed Cells appear; Differentiation of the TA cells (a), Differentiation of the stem cells (b).

7.5.4. Agent model

To deal with the agent modelling, we will use the program FLAME which generates the current state of the tissue. This is demonstrated in Fig. 7.5.4. We may see the growth of the cell colony at three time instants. It starts to grow from 6 stem cells (the darkest) and generates 2068 cells at the end of the process. The stem cells produce new stem cells, transit amplifying (TA) cells (fair) and the committed cells (mid-scale of colours). We may see the top layer of committed cells covering a layer of TA cells and several stem cells at the edge of the substrate at the end of the process.

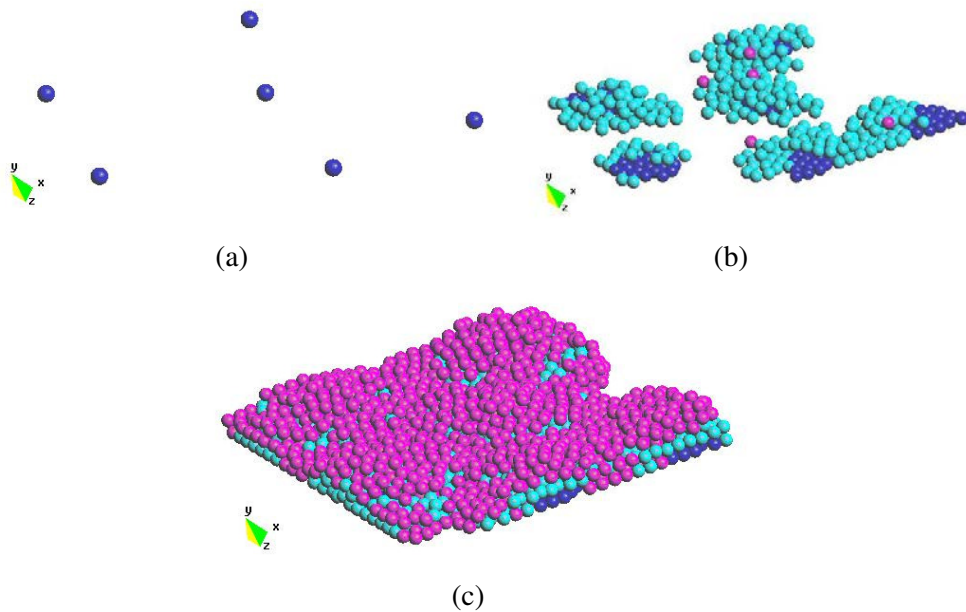


Figure 7.5.4. The colony formation; stem cells (a) clusters of cells around stem cells (b) layered structure of tissue (c).

7.5.5. Coupling scheme

The concept of the coupling scheme follows the Fig. 7.5.1 and is refined considering the functions of the agent and mechanical modules, Fig. 7.5.5.

```

For Each time step (T-module actions)
  For each agent
    Read state and position of neighbouring agents
    Update state and position as determined by internal rules and external signals
  Write new state and position of the agents
  End
  Update the mechanical model (its geometry)
  Transfer data about the state and position to the mechanical module
  Assign the appropriate material data to the cells
  The tasks of the mechanical module (M-module actions)
    Evaluate the state of displacements, strain and stress in the tissue including the
    cytoskeleton and membranes and the contact forces (mechanotransduction)

  Feedback to the agents – correct their positions

End
  
```

Figure 7.5.5. Refined coupling scheme.

The agents follow the rules about the differentiation, division and death depicted in the Subsec. 2.2.3 and illustrated in Fig. 7.5.4. We need to transfer the data about the actual state (in fact the kind) and position of the cells to the mechanical module. However, before we go into the FEM solver we need to update the geometry, discretization and material data of the model. The latter is due to the differentiation of the cells. Having the update we can solve the model for displacement, stresses and sensitivities. It is possible to transfer the resulting variables (displacements, stresses) into the agents and deal with the effects of the mentioned fields on the cell cycle.

7.5.6. Mechanical model

The mechanical model of a growing colony of cells should take into account the granularity of the medium that consists of a horde of objects. We think that a suitable mechanical model could be originated from discrete element method (DEM). The DEM defines a granular medium quite thoroughly [9–11]. An important feature of the DEM is the possibility of transferring of the stresses via contact forces. The feature quite intuitively helps to model mechanotransduction. In our model, the results that are obtained from the agent module stands for avatars of the cells. The avatars should be replaced with the physical models of particular cells. The replacement is done with the DEM particles. The results of such a replacement is shown in Fig. 7.5.6. The first approach was presented in [12]. We replaced the avatars with rigid particles that contained a simplified tensegrity cytoskeleton [13]. It was possible to transfer the forces via contact points between the particles [14, 15]. The scheme of such a particle is shown in Fig. 7.5.7. The icosahedron based cytoskeleton standing for the simplest single cell model is enriched with rigid membranes, Fig. 7.5.7b. However, it has appeared that the model is too crude. Then, the idea of application of a compliant particles is put into operation. The compliant model of a cell consists of cytoskeleton, cytoplasm, nucleus and cytoskeleton, Fig. 7.5.8a. The cell is fully discretized with finite elements. We have shown the displacement field and the shape of the cell during the AFM test, Fig. 7.5.8b. The equivalent strain field is shown in Fig. 7.5.8c. We do not discuss the results obtained from the model because it is under development.

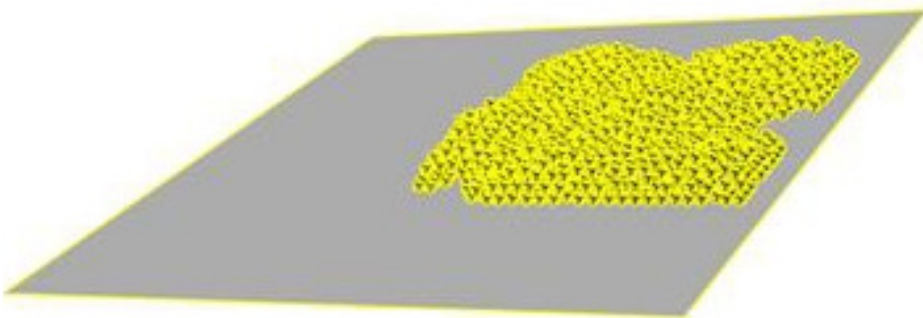


Figure 7.5.6. Avatars of the particles replaced with physical particles.

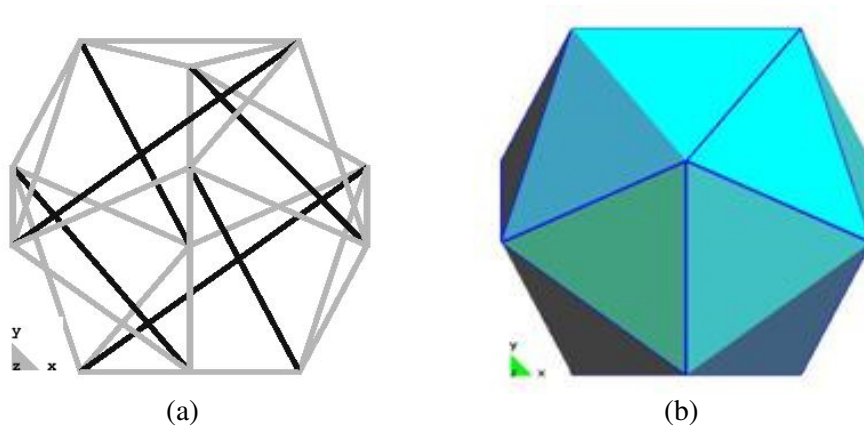


Figure 7.5.7. Tensegrity based model of cytoskeleton (a), the cytoskeleton enriched with membranes (b).

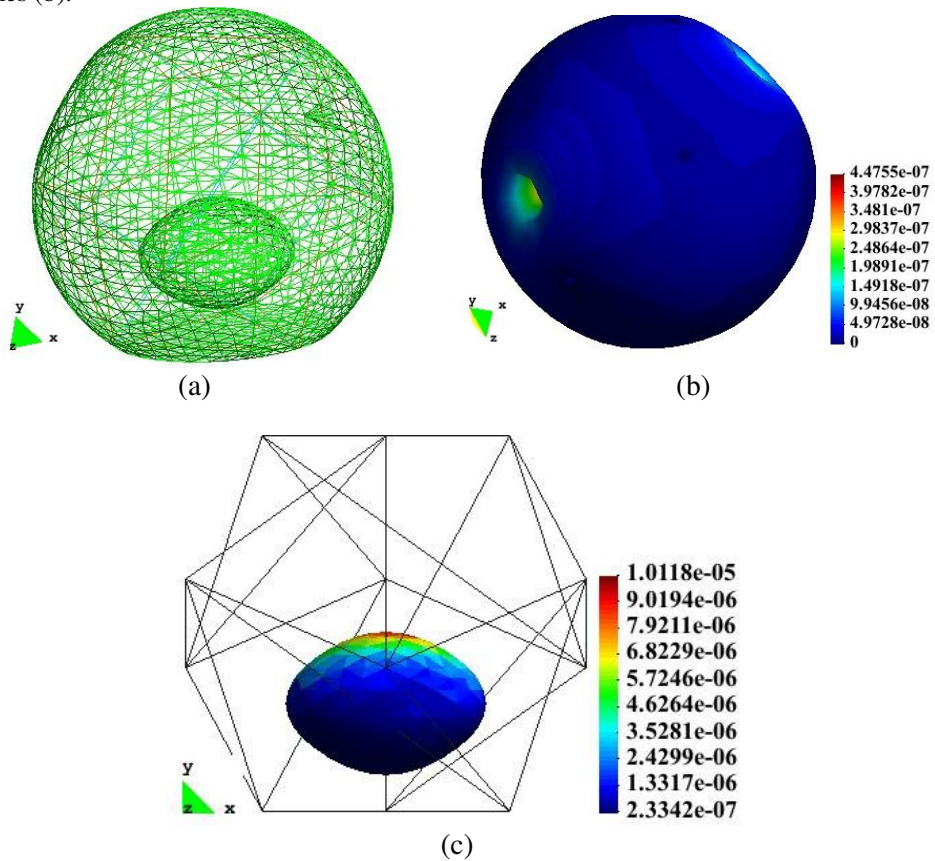


Figure 7.5.8. Scheme of a compliant single cell. The finite element mesh (a), displacement field (b), strains in the nucleus (c).

7.5.7. Summary

We have presented a concept of the development of an agent based model of a growing tissue. The novelties of the concept are as follows: the application of the staggered scheme to the agent-mechanical coupling, the application of the compliant particles with internal structure in DEM format.

References

- [1.] E. Postek. Concept of an Agent-stress Model of a Tissue. *Technische Mechanik*, 32(2–5):518–529, 2012.
- [2.] S. Adra, T. Sun, S. McNeil, M. Holcombe, and R. Smallwood. Development of a three dimensional multiscale computational model of the human epidermis. *PLoS ONE*, 5(1): e8511., 2010.
- [3.] T. Sun, P. McMinn, M. Holcombe, R. Smallwood, and S. MacNeil. Agent Based Modelling Helps in Understanding the Rules by Which Fibroblasts Support Keratinocyte Colony Formation. *PLoS ONE*, 3(5):e2129., 2008.
- [4.] C. Ainsworth. Stretching the imagination. *Nature*, 456:696–699, 2008.
- [5.] C. A. Felippa and K. C. Park. Staggered transient analysis procedures for coupled mechanical systems: formulation. *Comput. Methods Appl. Mech. Engrg.*, 24(1):61–111, 1980.
- [6.] E. Postek, R. W. Lewis, D. T. Gethin, and R. S. Ransing. Influence of initial stresses on the cast behaviour during squeeze forming processes. *Jou. of Mat. Proc. Tech.*, 159(3): 338–346, 2005.
- [7.] R. W. Lewis, E. Postek, Z. Q. Han, and D. T. Gethin. A finite element model of the squeeze casting processes. *Int. Jou. of Numer. Meth. for Heat and Fluid Flow*, 16(5): 539–572, 2006.
- [8.] S. F. Adra, S. Coakley, M. Kiran, and P. McMinn. An agent-based software platform for modelling systems biology. Epitheliome Project Report <http://www.flame.ac.uk>, 2008.
- [9.] P. Cundall and O. Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 29(1):47–65, 1979.
- [10.] J. Rojek, F. Zarate, C. de Saracibar, C. Gilbourne, and P. Verdoot. Discrete element modelling and simulation of sand mould manufacture for the lost foam process. *International Journal for Numerical Methods in Engineering*, 62(11):1421–1441, 2005.
- [11.] F. Dubois and M. Jean. The nonsmooth contact dynamic method: recent LMGC90 software developments and application. *Analysis and Simulation of Contact Problems Lecture Notes in Applied and Computational Mechanics*, eds P. Wriggers, U. Nackenhorst, 27:375–378, 2006.
- [12.] E. Postek, F. Dubois, and R. Mozul. Modeling of a collection of tensegrity particles with a nonsmooth discrete element method. *38th Solid Mechanics Conference*, Warsaw, Poland, 27–31.08.20, eds. R. Pecherski, J. Rojek, P. Kowalczyk, 28–29, 2012.
- [13.] D. Stamenovic. Effects of cytoskeletal prestress on cell rheological behavior. *Acta Biomater.* 1(3):255–262, 2005.

- [14.]E. Postek, R. Smallwood, and R. Hose. Nodal positions displacement sensitivity of an elementary icosahedral tensegrity structure. In *Computational Plasticity, Fundamentals and Applications*, E. Onate; D. R. J. Owen, eds., CD Proceedings of the *X International Conference on Computational Plasticity*, COMPLAS X, Barcelona, 2–4 September, 2009.
- [15.]E. Postek. Design Sensitivity of a Tensegrity Model of a Tissue. 19th International Conference on Computer Methods Mechanics, CD publication, 6 pages, 9–12 May, 2011. Warsaw, Poland.

7.6. Mould filling with solidification

Eligiusz Postek

7.6.1. Introduction

We present a model of mould filling process. The development of the model is presented in [1–2]. The incompressible Navier-Stokes equation is discretized with finite element method. The free surface tracking problem is solved using the pseudo-concentration function method. We follow the development of the solidification during mould filling. The thermal equation is discretized with finite element method as well. We present an industrial example to illustrate the features of the program.

7.6.2. Flow problem

The flow of material is assumed to be Newtonian and incompressible [3–5]. The governing Navier-Stokes equations are of the form:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = \nabla \cdot \mu \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T - \nabla p + \rho \mathbf{g} \right], \quad (7.6.1)$$

where \mathbf{u} is the velocity vector, p is the pressure, μ is the dynamic viscosity and \mathbf{g} is the gravitational acceleration vector. The mass conservation equation gives the incompressibility condition:

$$\nabla \cdot \mathbf{u} = 0. \quad (7.6.2)$$

The Galerkin procedure for the equations above is performed with a quadratic approximation for the velocities field, and linear approximation for the pressure field, it reads:

$$\mathbf{u} = \sum_{i=1}^{n_v} \mathbf{N} \mathbf{u}_i, \quad (7.6.3)$$

$$\mathbf{p} = \sum_{j=1}^{n_p} \mathbf{N}^j p_j . \quad (7.6.4)$$

The \mathbf{N} and \mathbf{N}^j denote the matrices of the approximation functions. The indices i and j correspond to velocity and pressure nodes, respectively.

The result of the discretization procedure is given by the Eq. (7.6.5):

$$\begin{bmatrix} \mathbf{M} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} - u^n \\ \Delta t \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{K}_u & \mathbf{Q} \\ \mathbf{Q}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u \\ 0 \end{bmatrix} . \quad (7.6.5)$$

The \mathbf{M} is the mass matrix; \mathbf{K} is the velocity stiffness matrix, \mathbf{Q} is the divergence matrix, and n is the time step. The right-hand side of the Eq. (7.6.5) contains external loading. The free surface is tracked with the volume of fluid method, [6]. The free surface is governed by the first order advection equation:

$$\frac{\partial F}{\partial t} + (\mathbf{u} \cdot \nabla) F = 0 . \quad (7.6.6)$$

The F is the pseudo-concentration function varying from -1 to 1. $F < 0$ indicates the empty region; $F > 0$ indicates the fluid region, $F = 0$ points out the free surface. The Eq. (7.6.6) is discretized with the Taylor-Galerkin method. An implicit time integration algorithm is used to solve (7.6.5), and when considering (7.6.6) an explicit integration scheme is used.

7.6.3. Microstructural solidification model

During the entire forming process, a part of the solidification takes place. In order to describe the process more accurately, a microstructure-based solidification model has been employed. The model stems from the assumptions given in [7–8]. The basic assumptions are as follows: the sum of the solid and liquid fractions is equal one, the solid fraction consists of dendritic and eutectic fractions:

$$f_l + f_s = 1 , \quad (7.6.7)$$

$$f_s = f_d + f_e . \quad (7.6.8)$$

Further assumptions are connected with the fact of the existence of interdendritic and intergranular eutectic fractions; the internal fraction consists of its dendritic and eutectic portions:

$$f_s = f_g^d f_i + f_g^e, \quad (7.6.9)$$

$$f_i = f_i^d + f_i^e. \quad (7.6.10)$$

The last equations lead to the final formulae for the dendritic and eutectic fractions with the assumption of the spherical growth:

$$f_d = f_g^d f_i, \quad (7.6.11)$$

$$f_e = f_g^d f_i^e + f_g^e, \quad (7.6.12)$$

$$f_g^d = \frac{4}{3} \pi N_d R_d^3, \quad (7.6.13)$$

$$f_i^e = \frac{4}{3} \pi N_e R_e^3. \quad (7.6.14)$$

The N_d , N_e are the grain densities, and R_d , R_e are the grain radii. The grain densities and grains sizes are governed by nucleation and growth evolution laws. The rate of growth of the dendritic and eutectic nuclei is given below. This depends on the undercooling, and a Gaussian distribution of the nuclei is assumed:

$$\dot{N}_{(d,e)} = N_{\max(d,e)} \frac{1}{2\pi} \exp\left(-\frac{\Delta T - \Delta T_{N(d,e)}}{2\Delta T_{\sigma(d,e)}}\right) \langle -\dot{T} \rangle. \quad (7.6.15)$$

$$\Delta T_{(d,e)} = T_{(d,e)} - T. \quad (7.6.16)$$

The rate of the dendritic and eutectic grain radii is established based on experimental dependence:

$$\dot{R}_{(d,e)} = f_{R(d,e)}. \quad (7.6.17)$$

Finally, the internal dendritic fraction depends on the melting temperature and k' is the partition coefficient:

$$f_i^d = 1 - \left(\frac{T_m - T}{T_m - T_l} \right)^{\frac{1}{k'-1}} \quad (7.6.18)$$

A numerical example concerning the mould filling follows.

7.6.4. Numerical example

Simulation of mould filling of an aluminium part (valve) employing the presented above solidification model is performed. The material of the mould is steel H13. The initial temperature of the cast is 650 deg and the 200 deg. Heat capacity and conductivity are the functions of temperature; radii rates (eutectic and dendritic) are given in Fig. 7.6.1 and 7.6.2.

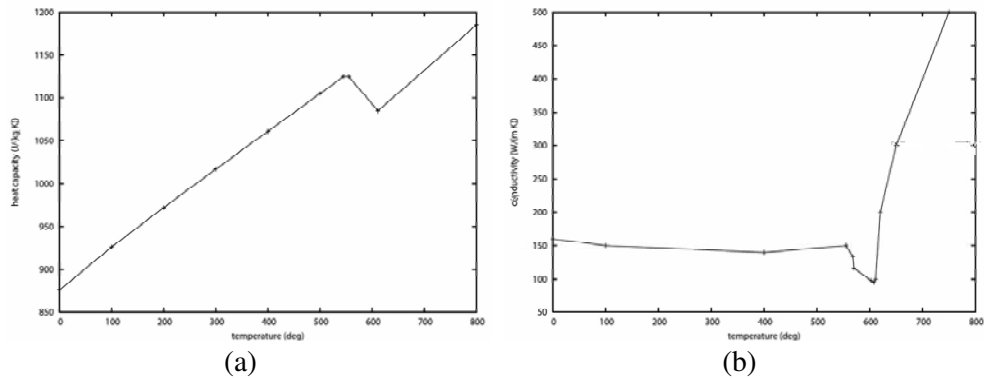


Figure 7.6.1. Heat capacity (a) and conductivity (b) versus temperature. eutectic and dendritic radius rates vs undercooling.

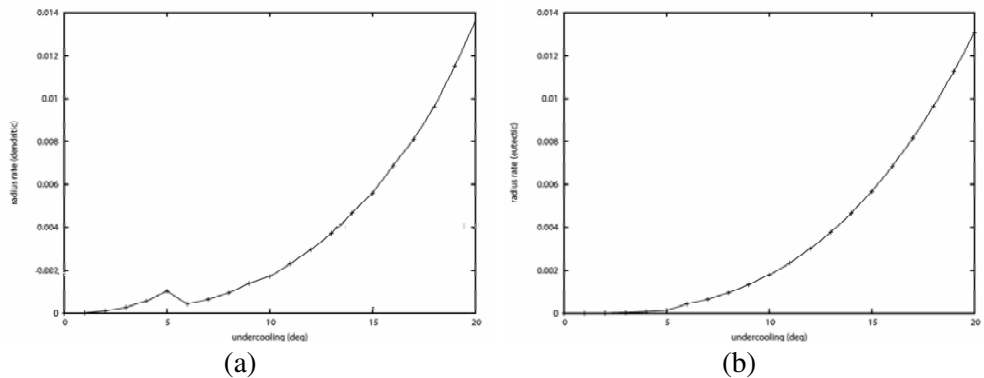


Figure 7.6.2. Dendritic (a) and eutectic (b) fractions radius rates versus undercooling.

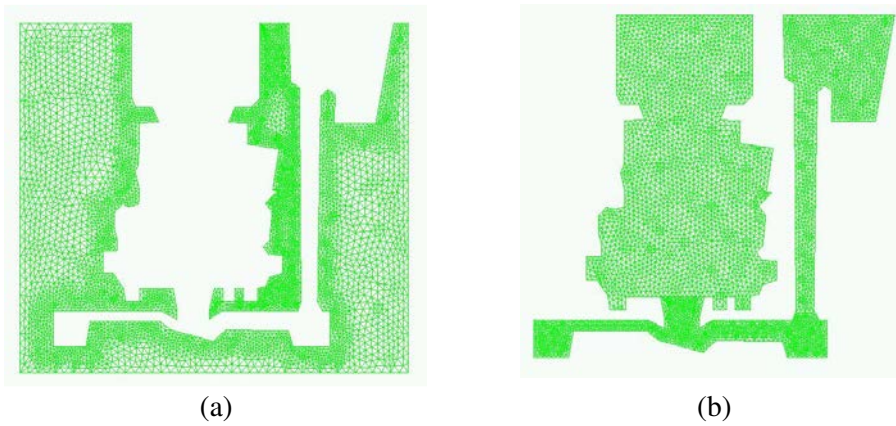


Figure 7.6.3. Finite element meshes of the mould (a) and the cast (b).

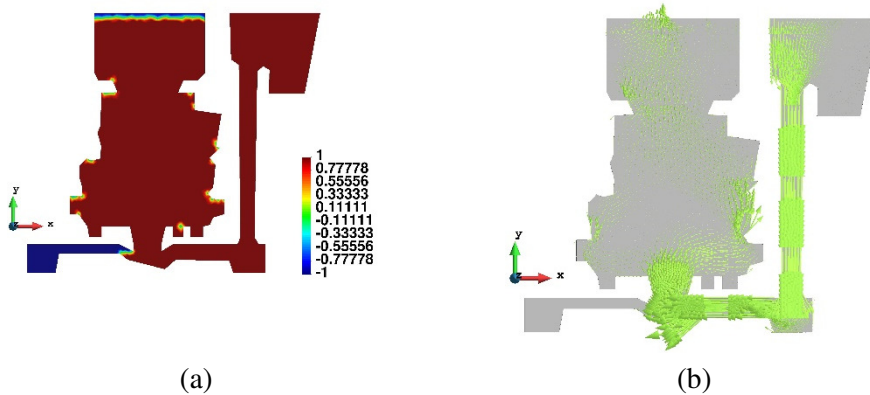


Figure 7.6.4. Pseudo-concentration function distribution (a), velocity field distribution (b).

The cast and mould are modelled using 10422 nodes. The meshes of the mould and cast are given in Fig. 7.6.3. The process of the mould filling is followed until the form is almost filled (95 sec.) that is also shown in Fig. 7.6.4a (distribution of the pseudo-concentration function). The left, lower branch of the mould is still unfilled. The temperature distribution and velocity field is presented in Fig. 7.6.5a.

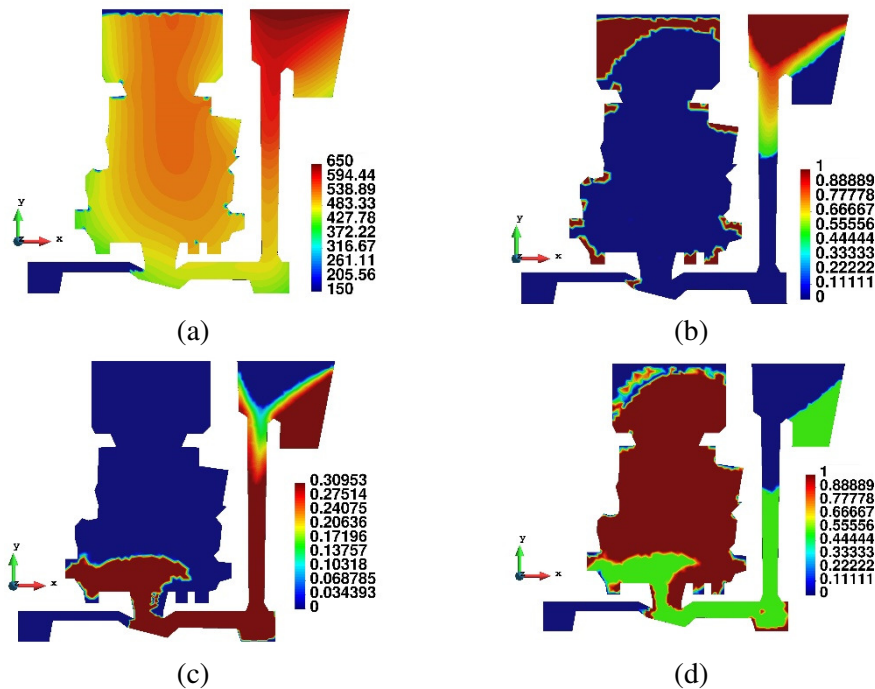


Figure 7.6.5. The state of the process at 95 sec: temperature distribution (a), liquidus distribution (b), dendritic fraction distribution (c), eutectic fraction distribution (d).

The distribution of the main microstructural variables, namely, distribution of the liquidus, dendritic and eutectic fractions is given in Fig. 7.6.5 b, c, d. It can be noticed that the dendritic fraction is concentrated in the thinner part of the section (inlet and lower part of the valve) while the eutectic one is concentrated in the main body of the part.

References

- [1.] R. W. Lewis, E. W. Postek, Z. Q. Han, and D. T. Gethin. A finite element model of the squeeze casting processes. *International Journal of Numerical Methods for Heat and Fluid Flow*, 16(5):539–572, 2006.
- [2.] E. W. Postek, R. W. Lewis, and D. T. Gethin. Finite element modelling of the squeeze casting process. *International Journal of Numerical Methods for Heat and Fluid Flow*, 18(3–4):325–355, 2008.
- [3.] C. Taylor and T. G. Hughes. *Finite element programming of the Navier Stokes equations*. Pineridge, Swansea, 1981.
- [4.] R. W. Lewis, K. Morgan, H. R. Thomas, and K. N. Seetharamu. *The finite element method in heat transfer analysis*. Wiley, 1996.
- [5.] K. Ravindran and R. W. Lewis. Finite element modelling of solidification effects in mould filling. *Finite Elements in Analysis and Design*, 31(2):99–116.
- [6.] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free surface boundaries. *Journal of Computational Physics*, 39(1):210–25, 1981.

- [7.] Ph. Thevoz, J. Desbiolles, and M. Rappaz. Modelling of equiaxed microstructure formation in casting. *Metall. Trans. A*, 20(2):311–322, 1989.
- [8.] D. J. Celentano. A thermomechanical model with microstructure evolution for aluminium alloy casting processes. *Int. J. of Plasticity*, 18(10):1291–1335, 2002.

Appendix

There is large number of original computational programs developed at IPPT. Some of them are based on commercial codes, other present complete new research code dedicated to solve specialized tasks. Below we list only some of them hoping that this list will be useful for research colleagues from the Institute and external institutions.

A.1. FibFlex – Computation of flexible polymer nanofibers bending

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Fluid Physics and Mechanics

Category

Simulation, data analysis, image processing

Downloads and service

Krzysztof Zembrzycki: kzem@ippt.pan.pl
Patryk Hejduk: patrykhejduk@gmail.com

Description

Flexible polymer nanofibers computation program has been developed in Department of Fluid Physics and Mechanics. Main goal of this program is to find exact positions of all the segments of flexible nanofibers from series of pictures. It is capable of detection in 2 or 3 dimensions. Program calculates displacements of indicated points, caused by fluid flow and Brownian motion, measuring bending properties by fitting an appropriate curve to an processed image with extracted fibre shape. Due to complex multidimensional shape recognition and amount of data it is run on Grafen HPC, in future for even faster analysis it could be re-written to process on GPU units. Program is written in MATLAB programming language.

References

- [1.] S. Pawłowska, P. Nakielski, P. Hejduk, F. Pierini, K. Zembrzycki, and T. A. Kowalewski. Brownian motion of nanofibres. *Book of Abstracts*, 162–163, 4th National Conference of Nano and Micromechanics, Wrocław 2014.

A.2. SpatKin – rule-based modeling of spatially extended biochemical systems

Institution

Institute of Fundamental Technological Research (IPPT),
Laboratory of Modeling in Biology and Medicine

Category

Computer simulations of stochastic processes

Downloads and service

Source code is available from the repository <http://spatkin.googlecode.com/svn>.
Marek Kočańczyk: mkochan@ippt.pan.pl

Description

Continuous approximations, such as ODE or PDE, are not appropriate for the simulation of the dynamics of membrane signaling systems, as the low number of reacting molecules introduces significant noise that leads to stochastic effects. Also, slow diffusion in the membrane discards approaches which assume the immediate communication of all reagents in the system, such as the Gillespie algorithm, that is incapable of simulating spatial effects. Cell signaling systems involve proteins that can assume multiple possible states resulting from dynamic post-translational modifications and formation of transient complexes. Networks of molecular interactions are, therefore, inherently characterized by the combinatorial complexity. In many cases, the interaction network can be more concisely defined by a set of rules, which define interactions between protein domains, rather than between proteins or protein complexes. In order to be able to simulate and study dynamics of such complex, stochastic, spatially extended systems, the software called SpatKin has been developed. Applied approach ensures the exact state-to-state dynamics of the underlying Markov process: reaction and diffusion events are selected from the catalog of possible events (determined by

rules) and are fired with their propensities proportional to corresponding rate constants. The catalog is always complete as, after performing any reaction, it is updated by considering every possible new event that may happen in the updated system. Such local updates are feasible because space is discretized into hexagonal cells: a cell may contain a single protein or protein complex; in one step molecules can move to adjacent empty cells and bimolecular reactions can occur between molecules that are placed in the neighboring cells. Contrary to similar conventional modeling approaches, the full network of possible interactions is not generated according to the rules – instead, the network is evaluated locally for existing molecular species that reduces the overall computational complexity of the algorithm.

References

- [1.] M. Kočańczyk, J. Jaruszewicz, and T. Lipniacki. Stochastic transitions in a bistable reaction system on the membrane. *J R Soc Interface*, 10 (84):20130151, 2013.
- [2.] P. J. Żuk, M. Kočańczyk, J. Jaruszewicz W. Bednorz, and T. Lipniacki. Dynamics of a stochastic spatially extended system predicted by comparing deterministic and stochastic attractors of the corresponding birth-death process. *Physical Biology*, 9(055002), 2012.

A.3. DrugRelease

Numerical Modeling of Drug Release from Nanofibers

Institution

Institute of Fundamental Technological Research (IPPT),
Laboratory of Modeling in Biology and Medicine

Category

BioMed

Downloads and service

Paweł Nakielski: pnakiel@ippt.pan.pl

Description

COMSOL Multiphysics®, Numerical Modeling of Drug Release from Nanofibers is a commercial software based on the finite element method with the module allowing to implement author's mathematical model with the appropriate boundary and initial conditions. Comprehensive studies of drug release from nanofibres based materials have been performed to predict drug release profiles. In the numerical study, we constructed a 3D geometry representing nanofibrous cubic element (Fig. 9.3.1). Drug release from nanofibers was modeled by adsorption-desorption and diffusion equation. Analysis of the impact of fibers arrangement, release kinetic, and material parameters were presented in several papers. Our study shows that reducing the distance between the fibers and increasing the regularity of the structure leads to slower drug release from the material. On the other hand, increase of the porosity can enhance the release rate and cause adverse effect that is named burst release.

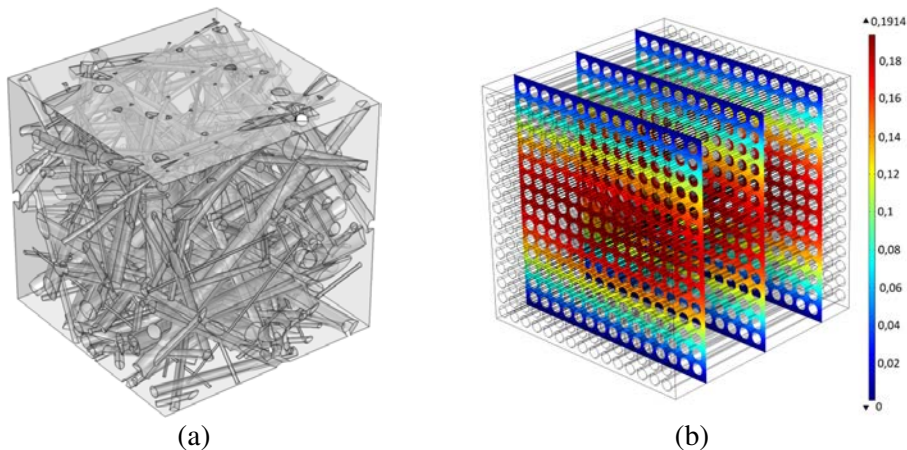


Figure 9.3.1. (a) Geometry of randomly oriented fibers. Porosity of the material equals $\epsilon = 0.72$ (b). Simulated drug concentration distribution in fluid between regular fibers after 14 days of release. Diameter of the fibers equals $r = 2.0 \mu\text{m}$.

Links

Comsol Multiphysics®, Comsol Inc., www.comsol.com

References

- [1.] P. Nakielski, T. Kowalczyk, K. Zembrzycki, and T. A. Kowalewski. Experimental and numerical evaluation of drug release from nanofiber mats to brain tissue. *J. Biomed. Mater. Res. Part B Appl. Biomater.*, 103B, 282–291, 2015.
- [2.] P. Nakielski. Symulacje numeryczne procesu desorpcji i dyfuzji leków w materiale z nanowłókien. *Model. inżynierskie* 48:9–111, 2013.
- [3.] P. Nakielski, T. Kowalczyk, and T.A. Kowalewski. Modeling Drug Release from Materials Based on Electrospun Nanofibers. Full-paper in Proc. COMSOL Conf. Rotterdam 23–25 October 2013.

A.4. AceGen/AceFEM

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Mechanics of Materials

Category

Finite element method, code generation system

Downloads and service

Stanisław Stupkiewicz: sstupkie@ippt.pan.pl

Description

AceGen is a hybrid symbolic-numerical system for the derivation and automatic generation of computer codes. AceGen is an add-on package to Mathematica and combines symbolic capabilities of Mathematica with an automatic differentiation (AD) technique and code optimisation technique. AceFEM is a flexible, general purpose finite element code. AceFEM is closely integrated with AceGen that enables highly efficient and robust implementation of complex material models, as well as development and testing of new formulations for a variety of problems in computational mechanics. AceGen and AceFEM have been developed by Prof. J. Korelc (University of Ljubljana, Slovenia).

The group of Prof. S. Stupkiewicz at IPPT has contributed to the development of AceFEM by implementing a contact environment. AceGen/AceFEM is also exten-

sively used in the research on various topics that include: computational contact mechanics, wear, hydrodynamic lubrication, computational plasticity, development of advanced material model and others.

Links

AceGen/AceFEM web site: <http://www.fgg.uni-lj.si/Symech/>

References

- [1.] J. Lengiewicz, J. Korelc, and S. Stupkiewicz, Automation of finite element formulations for large deformation contact problems. *International Journal for Numerical Methods in Engineering*, 85(10):1252–1279, 2011.
- [2.] J. Lengiewicz and S. Stupkiewicz. Continuum framework for finite element modelling of finite wear. *Computer Methods in Applied Mechanics and Engineering*, 205-208:178–188, 2012.
- [3.] S. Stupkiewicz and H. Petryk. A robust model of pseudoelasticity in shape memory alloys. *International Journal for Numerical Methods in Engineering*, 93(7):747–769, 2013.
- [4.] S. Stupkiewicz, A. Piccolroaz, and D. Bigoni. Elastoplastic coupling to model cold ceramic powder compaction. *Journal of the European Ceramic Society*, 34:2839–2848, 2014.
- [5.] J. Lengiewicz, M. Wichrowski, and S. Stupkiewicz. Mixed formulation and finite element treatment of the mass-conserving cavitation model. *Tribology International*, 72:143–155, 2014.
- [6.] J. Korelc and S. Stupkiewicz. Closed-form matrix exponential and its application in finite-strain plasticity. *International Journal for Numerical Methods in Engineering*, 98:960–987, 2014.

A.5. Crack modeling in composites package

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Mechanics of Materials, Advanced Composite Materials Group

Category

Fracture analysis and effective material properties of composites.

Downloads and service

Michał Basista: mbasista@ippt.pan.pl

Description

The Crack Modeling in Composites Package has been developed in IPPT PAN in Advanced Composite Materials Group. Main features of the package are as follows:

- Composite microstructure analysis via micro-CT scans with the use of Simpleware ScanIP+FE software,
- FEM mesh generation of the real composite microstructure for ABAQUS software with the use of Simpleware ScanIP+FE software,
- Calculations of the fracture toughness of the real composite materials with the use of ABAQUS software,
- Estimations of the effective material properties of the real composite materials with the use of ABAQUS software.

The package uses Intel Fortran software and subroutines written in Fortran. Calculations are performed with the use of ABAQUS software. Some of the calculations may be performed also with the use of FEAP program.

References

- [1.] Z. Poniżnik, V. Salit, M. Basista, and D. Gross. Effective elastic properties of interpenetrating phase composites. *Computational Materials Science*, 44(2): 813–820, 2008.
- [2.] Z. Poniżnik, Z. Nowak, and M. Basista. Comparative analysis of different numerical models of fracture of single fibre in ceramic matrix (in preparation).

A.6. Residual Thermal Stress (RTS) Package

Institution

Institute of Fundamental Technological Research (IPPT),
Mechanics of Materials, Advanced Composite Materials Group

Category

Material properties and residual stress analysis.

Downloads and service

Michał Basista: mbasista@ippt.pan.pl

Description

The RTS package has been developed at IPPT PAN in Advanced Composite Materials Group. Main features of the package are:

- It uses real microstructure from Computed Tomography (CT) images for building FE mesh. The ScanIP/Fe software is used for generation of FE mesh,
- Using the developed package thermal residual stresses generated after sintering process can be predicted with reasonable accuracy,
- Part of the developed package can also be used for prediction of elastic and thermal properties of composites with account of real material microstructure,
- Effect of RTS-induced damage is included in the package.

The calculations can be performed in Abaqus (or FEAP).

References

- [1.] W. Węglewski, K. Bochenek, M. Basista, Th. Schubert, U. Jehring, J. Litniewski, and S. Mackiewicz. Comparative assessment of Young's modulus measurements of metal-ceramic composites using mechanical and non-destructive tests and micro-CT based computational modelling. *Computational Material Sciences*, 77:19–30, 2013.
- [2.] W. Węglewski, M. Basista, M. Chmielewski, and K. Pietrzak. Modeling of thermally induced damage in the processing of Cr-Al₂O₃ composites. *Composites Part B: Engineering*, 43(2):255–264, 2012.

A.7. YADE-PM – Extension of YADE discrete element method environment

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Mechanics of Materials

Category

Discrete Element Method analysis, analysis of advanced meta-materials.

Downloads and service

Paweł Hołobut: pholob@ippt.pan.pl

Description

YADE-PM is the extension of YADE – the open-source discrete element method environment. The main purpose of the extension is to effectively analyse the behaviour of a special class of active multi-modular structures needed for the purpose of the developed future-technology called Programmable Matter. The modules in the structure, besides purely passive mechanical interactions, are capable of actively exert and control inter-modular forces and torques, connect and disconnect, as well as can communicate with neighbours.

YADE-PM is written in C++ and Python, with QT library for the purpose of visualisation. It is available for Linux and Windows OS.

Acknowledgment:

This work has been partially supported by the project “Micromechanics of Programmable Matter” (contract no. UMO–2011/03/D/ST8/04089 with the National Science Centre (NCN) in Poland).

Links

YADE web-page: <http://yade-dem.org/>

References

- [1.] P. Hołobut, M. Kurasa, and J. Lengiewicz. A class of evolving microstructures for scalable collective actuation of Programmable Matter. *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (accepted), 2014.

A.8. Nanotrack: Nanoparticles motion detection program

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Fluid Physics and Mechanics

Category

Simulation, data analysis, image processing.

Downloads and service

Krzysztof Zembrzycki: kzem@ippt.pan.pl
Tomasz A. Kowalewski: tkowale@ippt.pan.pl

Description

Nanoparticles' motion detection program has been developed in IPPT PAN in Department of Fluid Physics and Mechanics. Program is capable of simulating brownian motion diffusion with different initial conditions either in 2 or 3 dimensions. Main feature of our system is recognition and positioning particles in nanoscale on pictures from a high-speed camera in very low light conditions. It could be applied in near wall fluid flow measurements, diffusion of nanorods in fluids, calibration optical trap apparatus. Program is written in MATLAB programming language. Due to the large amount of data and complex particle detection tools it is run on Grafen computer cluster.

References

- [1.] K. Zembrzycki, S. Błoński, and T. A. Kowalewski. Analysis of wall effect on the process of diffusion of nanoparticles in a microchannel. *J. Phys.: Conf. Ser.* 392:012014, 2012.

A.9. Original computer program MC-UNBAL

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Intelligent Technologies

Category

Numerical simulations for vibrating rotor shaft.

Downloads and service

Piotr Tazowski: ptazow@ippt.pan.pl

Description

The computer program MC-UNBAL is an originally developed numerical code for simulations of steady state dynamic responses of the rotor-shaft-bearing systems excited by residual static and dynamic unbalances. This program is based on the structural mechanical model taking into account full geometry and material constants of the rotor-shaft as well as inertial-visco-elastic properties of the bearing supports.

Program is written in FORTRAN. It allows for easy multiplatform compilation, therefore MC-UNBAL is available on Windows as well as on Linux operating system.

References

- [1.] R. Stocki, T. Szolc, P. Tuzowski, and J. Knabel. Robust design optimization of the vibrating rotor shaft system subjected to selected dynamic constraints. *Mechanical Systems and Signal Processing*, vol. 29, p. 34–44, 2012.
- [2.] R. Stocki, R. Lasota, P. Tuzowski, and T. Szolc. Scatter assessment of rotating system vibrations due to uncertain residual unbalances and bearing properties. *Computer Assisted Mechanics and Engineering Sciences*, vol. 19, p. 95–120, 2012.

A.10. STAND – STochastic ANalysis and Design

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Computational Science

Category

Optimization, statistical, and reliability analysis.

Downloads and service

Piotr Tuzowski: ptuzow@ippt.pan.pl

Description

The STAND System has been developed in IPPT PAN in Department of Computational Science. It is reliability, optimization and stochastic analysis system. Main features of the program are:

- user friendly graphical user interface allowing comfortable task definition as well as results post processing,
- several types of deterministic and robust optimization such as: Nelder-Mead Simplex, simulated annealing, random search. dual response surface,
- reliability analysis such as : FORM, SORM, Importance Sampling (including adaptive version), Monte Carlo,
- response surface analysis - allows for derivation of analytical expression for given function estimation (first order, second order and kriging).

Program is written in C++ with support of Qt library. It allows for easy multiplatform compilation, therefore STAND is available on Windows as well as on Linux operating system.

References

- [1.] R. Stocki, K. Kolanek, J. Knabel, and P. Tuzowski. FE based structural reliability analysis using STAND environment. *Computer Assisted Mechanics and Engineering Sciences*, vol. 16, p. 35–58, 2009.
- [2.] R. Stocki, T. Szolc, P. Tuzowski, and J. Knabel. Robust design optimization of the vibrating rotor shaft system subjected to selected dynamic constraints. *Mechanical Systems and Signal Processing*, vol. 29, p. 34–44, 2012.

A.11. DCAST

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Computational Science

Category

Coupled problems, thermomechanics, casting processes.

Downloads and service

Eligiusz Postek: epostek@ippt.pan.pl

Description

The DCAST is a program dedicated to transient thermo-mechanical problems. This is mostly oriented on simulations of the casting processes. The mechanical problem can be viscous-elastic. The most important features of the thermal problem are the analysis of solidification and the analysis of microstructural solidification. The gap width between the cast and the mould is included into the analysis.

References

- [1.] E. Postek, R. W. Lewis, D. T. Gethin, and R. S. Ransing. Influence of initial stresses on the cast behaviour during squeeze forming processes. *Journal of Materials Processing Technology*, 159(3):338–346, 2005.
- [2.] R. W. Lewis, E. Postek, Z.Q. Han, and D. T. Gethin. A finite element model of the squeeze casting processes. *International Journal of Numerical Methods for Heat and Fluid Flow*, 16(5):539–572, 2006.

A.12. DMFILL

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Computational Science

Category

Coupled problems, thermomechanics, casting processes

Downloads and service

Eligiusz Postek: epostek@ippt.pan.pl

Description

The DMFILL is a program dedicated to transient thermo-fluid problems. This is mostly oriented on simulations of the mould filling processes. It uses volume of fluid

(VOF) front tracking procedure. The important feature of the thermal problem is the analysis of microstructural solidification during mould filling process.

References

- [1.] R. W. Lewis, E. Postek, Z.Q . Han, and D. T. Gethin. A finite element model of the squeeze casting processes. *International Journal of Numerical Methods for Heat and Fluid Flow*, 16(5):539–572, 2006.
- [2.] E. Postek, R. W. Lewis, and D. T. Gethin. Finite element modelling of the squeeze casting process. *International Journal of Numerical Methods for Heat and Fluid Flow*, 18(3-4):325–355, 2008.

A.13. DEMSYS

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Computational Science

Category

Discrete Element Method, geomechanics, bulk materials, composites.

Downloads and service

Jerzy Rojek: jrojek@ippt.pan.pl

Description

The DEMSYS is a Discrete Element Method program with numerous constitutive models of the rigid particles interactions. The particles can interact with compliant bodies. The bodies can be discretized with finite elements. The equation of motion is integrated explicitly. The program has a modern modular structure that exploits the features of the fortran 90/95. The pre- and the postprocessing are adjusted to the GiD program.

Links

<http://www.gidhome.com/>

References

- [1.] S. Nosewicz, J. Rojek, K. Pietrzak, and M. Chmielewski. Viscoelastic discrete element model of powder sintering. *Powder Technology*, 246:157–168, 2013.
- [2.] J. Rojek, G. F. Karlis, L.J. Malinowski, and G. Beer. Setting up virgin stress conditions in discrete element models. *Computers and Geotechnics*, 48:228–248, 2013.
- [3.] J. Rojek C. Labra, O. Su, et al. Comparative study of different discrete element models and evaluation of equivalent micromechanical parameters. *International Journal of Solids And Structures*, 49(13):1497–1517, 2012.

A.14. DTENS

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Computational Science

Category

Tensegrity structures, parameter sensitivity, cell biology.

Downloads and service

Eligiusz Postek: epostek@ippt.pan.pl

Description

The DTENS is a program dedicated to the analysis of tensegrity structures. In particular, it is possible to simulate changes in the displacements, strains and stress fields during adding or removal of the particular cells. The single cell is modelled as a tensegrity structure with equivalent microtubules and actins. One of the feature of the program is the parallel implementation of the Direct Differentiation Method (DDM) of the parameter sensitivity algorithm.

References

- [1.] E. Postek, R. Smallwood, and R. Hose. Nodal positions displacement sensitivity of an elementary icosahedral tensegrity structure, (CD publication, 6 pages), X International Conference on Computational Plasticity, eds. E. Onate, D.R.J. Owen, B. Suarez, COMPLASX, Barcelona, Spain, 2–4th September, 2009.

- [2.] E. Postek, R. Smallwood, and R. Hose. Behaviour of tensegrity cells assembly during single cell growth, In 1st International Conference on Computational & Mathematical Biomedical Engineering, eds. P. Nithiarasu, R. Lohner, 415–418, Swansea, UK, June 29th–July 1, 2009.
- [3.] E. Postek. Concept of an Agent-stress Model of a Tissue. *Technische Mechanik*, 32:518–529, 2012.

A.15. OREGANO-VE

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Computational Science

Category

Tectonics, creep, seismic cycle, Rayleigh-Taylor instability.

Downloads and service

Eligiusz Postek: epostek@ippt.pan.pl

Description

The Oregano-VE is a program dedicated to tectonophysics problems. It solves problems of quasi-static viscous flows and viscoelastic problems. The applications cover stability of the crust and problems of seismic cycle, dike intrusion, plate collisions, etc.

Links

<http://www.see.leeds.ac.uk/people/g.houseman>

References

- [1.] T. Hoogenboom and G. A. Houseman. Rayleigh-Taylor instability as a mechanism for corona formation on Venus. *ICARUS*, 180:292–307, 2006.
- [2.] B.A.R. Youngs and G. A. Houseman. Formation of steep-sided topography from compositionally distinct dense material at the base of the mantle. *Journal of Geophysical Research B: Solid Earth*, 114:(B04404), 2009.

- [3.] E. Postek, G. A. Houseman, and P. K. Jimack. The effect of geometrical nonlinearity in visco-elastic deformation. European Geophysical Union General Assembly, 2007.
- [4.] E. Postek, G. A. Houseman, and P. K. Jimack. Nonlinear effects during post-seismic visco-elastic deformation, JGS003 - Earthquake and Volcano Geodesy. International Union of Geodesy and Geophysics, XXIV General Assembly, Peruggia, Italy, July 2–13th, 2007.
- [5.] E. Postek, G. A. Houseman, and P. K. Jimack. Surface deformations during the earthquake cycle for a layered visco-elastic crust. 10th International Workshop on Modeling of Mantle Convection, Carry-le-Rouet, France, 2007.
- [6.] E. Postek, G. A. Houseman, and P. K. Jimack. Generic Models of Linear and Non-linear Visco-elastic Surface Deformation above a Fault, EGU2008-A-04783, GD8-1TH2P-0093, Session GD8: Modelling and Monitoring the Deformation and State of Stress of the Lithosphere (co-sponsored by the International Lithosphere Program Task Force VII), European Geophysical Union General Assembly, Vienna, 13–18 April, 2008.

A.16. VECDS – Visual Editor of Crystal Defects the program for preprocessing of atomistic models

Institution

Institute of Fundamental Technological Research (IPPT),
Department of Computational Science

Category

Computational mechanics and physics, preprocessing of atomistic models of
crystal defects

Downloads and service

The program is available from the site <http://vecds.sourceforge.net>. Information
about the code, licence, developers can be found on this web site.

Paweł Dłużewski: pdluzew@ippt.pan.pl

Description

The VECDS program is developed in IPPT PAN since 2005 under direction of
Pawel Dluzewski in the framework of the following projects: PARSEM (6FP MRTN-

CT-2004-005583), R15 012 03 and NCN N N519 647640. The program consist of two main parts. The first part concerns the computational segments:

- generation of a perfect structure or heterostructure like Cu/Al₂O₃, InGaN/GaN, C/SiC and others,
- rotation of the heterostructure on the basis of given Miller coefficients or Euler angles,
- calculation of displacement fields induced by elemental dislocations, stacking faults and other extended defects considered,
- subsequent or simultaneous input of all dislocation lines (with subsequent jumps) into the heterostructure,
- computer processing of resultant mesh in order to satisfy symmetry conditions imposed on the resultant periodicity cell, as a whole [1, 2].

The second part consists of a 3D graphical user interface thanks to which the structure can be rotated and the dislocations can be input by using the mouse.

References

- [1.] J. Cholewiński, M. Maździarz, G. Jurczak, and P. Dłużewski. *Int. J. Multiscale Comput. Engng.*, 2014 (in print).
- [2.] P. Dłużewski and J. Cholewiński. *Annual Report 2013*, Polish Academy of Sciences, pages 80–82, 20