BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF CIVIL ENGINEERING
FAKULTA STAVEBNÍ

INSTITUTE OF STRUCTURAL MECHANICS
ÚSTAV STAVEBNÍ MECHANIKY

THE EXPLOITATION OF PARALLELIZATION TO NUMERICAL SOLUTIONS REGARDING PROBLEMS IN NONLINEAR DYNAMICS
VYUŽITÍ PARALELIZACE PŘI NUMERICKÉM ŘEŠENÍ ÚLOH NELINEÁRNÍ DYNAMIKY

DOCTORAL THESIS STATEMENT
TEZE DIZERTAČNÍ PRÁCE

AUTHOR — Ing. Václav Rek
AUTOR PRÁCE

SUPERVISOR — doc. Ing. Ivan Němec, CSc.
VEDOUCÍ PRÁCE

BRNO 2018

Abstract
The main aim of this thesis is the exploration of the potential use of the parallelism of numerical computations in the field of nonlinear dynamics. In the last decade the dramatic onset of multicore and multi-processor systems in combination with the possibilities which now provide modern computer networks has risen. The complexity and size of the investigated models are constantly increasing due to the high computational complexity of computational tasks in dynamics and statics of structures, mainly because of the nonlinear character of the solved models. Any possibility to speed up such calculation procedures is more than desirable. This is a relatively new branch of science, therefore specific algorithms and parallel implementation are still in the stage of research and development which is attributed to the latest advances in computer hardware, which is growing rapidly. More questions are raised on how best to utilize the available computing power. The proposed parallel model is based on the explicit form of the finite element method, which naturaly provides the possibility of efficient parallelization. The possibilities of multicore processors, as well as parallel hybrid model combining both the possibilities of multicore processors, and the form of the parallelism on a computer network are investigated. The designed approaches are then examined in addressing of the numerical analysis regarding contact/impact phenomena of shell structures.

Abstrakt
Hlavním cílem této práce je prozkoumání možností využití paralelizace v numerických výpočtech nelineární dynamiky. V poslední dekádě došlo k dramatickému nástupu vícejádrových a výceprocesorových systému v kombinaci s možnostmi, které nyní poskytují moderní počítačové sítě. Komplexnost a velikost řešených modelů se neustále zvyšuje a díky vysoké výpočetní náročnosti úloh dynamiky a statiky konstrukcí, a to především kvůli jejich často nelineárnímu charakteru, je jakákoliv možnost urychlení výpočetních procedur více než žádoucí. Jelikož se jedná o relativně nové odvětví, řada algoritmů a konkrétních paralelních implementací je stále ve stádiu vývoje a výzkumu, a to i proto, že pokroky v oblasti počítačového hardwaru rapidně vzrůstají a s tím vznikají další otázky, jak nejlépe využít dostupný výpočetní výkon. Navržený paralelní model je založený na explicitní formě metodě konečných prvků, která ze své podstaty poskytuje možnost efektivní paralelizace. Zkoumány jsou pak možnosti využití vícejádrovch procesorů, ale i hybridního paralelního modelu kombinujícího možnosti vícejádrových procesorů a paralelní formy na počítačové síti. Navržené přístupy jsou pak testovány při numerickém řešení kontaktní/impaktní úlohy skořepinových konstrukcí.

Keywords
FEM, Explicit Form of FEM, FDM, Dynamics of Structures, Parallel Computing, GPGPU, NVIDIA CUDA, Computer Network, TCP/IP, .NET, C/C++, C#

Klíčová slova
MKP, explicitní forma MKP, MKD, dynamika konstrukcí, paralelní výpočty, GPGPU, NVIDIA CUDA, počítačové sítě, TCP/IP, .NET, C/C++, C#

REK, Václav. *The Exploitation of Parallelization to Numerical Solutions Regarding Problems in Nonlinear Dynamics*. Brno, 2018. 223 p. Doctoral thesis. Brno University of Technology. Faculty of Civil Engineering. Supervisor Ivan NĚMEC.

Brno University of Technology
Faculty of Civil Engineering
Institute of Structural Mechanics

The Exploitation of Parallelization to Numerical Solutions Regarding Problems in Nonlinear Dynamics

by

Václav Rek

Doctoral degree study program: Structures and Traffic Constructions

Doctoral thesis statement for obtaining
the academic title of "Doctor" abbreviated to "Ph.D."

Brno, July 2018

Ph.D. Candidate: Václav Rek
Institute of Structural Mechanics
Faculty of Civil Engineering
Brno University of Technology
Veveří 331/95, 602 00 Brno, Czech Republic
rekv@seznam.cz

Supervisor: Ivan Němec
Institute of Structural Mechanics
Faculty of Civil Engineering
Brno University of Technology
Veveří 331/95, 602 00 Brno, Czech Republic
nemec.i@fce.vutbr.cz

Reviewers: _____

_____

_____

The doctoral thesis is deposited at the Faculty of Civil Engineering, Brno University of Technology.

ISBN: _____

ISSN: _____

Contents

1 Introduction and Motivation

1.1 Introduction

Thanks to rapid advances during the last decade in computer technology, in the field of multicore and multiprocessor technology, a lot of attention has been devoted to the parallel processing of data in many scientific and industrial sectors. This is in great contrast to previous decades when an increase in computing power, especially in personal computers, was achieved by increasing the clock speed of processors and has resulted in increase in energy consumption and thus energy inefficiency. This type of technological evolution has its limitations compared to multi-processors and multi-core computer architectures respectively. Multi-processor platforms are the oldest way how to distribute time intensive computations.

A technology which is completely separate from the previously mentioned technologies is the technology of quantum computers which promises to solve problems that are intractable on digital computers.

Now it is quite common to use a CPU (Central Processing Unit) for the parallel run of computational time-consuming tasks in synchronous and asynchronous order. Parallel programming is especially supported by the software libraries OpenMP (Open Multi-Processing, the oldest multi-platform shared memory multiprocessing programming based on compiler's special directives, see `https://msdn.microsoft.com/en-us//library/tt15eb9t.aspx`),
MPI (Message Passing Interface, developed and maintained by a consortium of academic, research, and industry partners, see `http://www.mpi-forum.org/`, [36]), API (Application Programming Interface) of UNIX-like operating systems (POSIX - Portable Operating System Interface, see `http://standards.ieee.org/develop/wg/POSIX.html`), Microsoft Windows (Windows API, see `https://msdn.microsoft.com/en-us/library/windows/desktop/ff818516(v=vs.85).aspx`, [42]) operating systems or the other OS. Nowadays it is also possible to use native threads in C++ standard libraries (since its version 11, see `https://isocpp.org/wiki/faq/cpp11#cpp11-what`). The aforementioned technlogies are focused mainly on usage of C and C++ programming language. In some form they are also accessible even for the historically most widely used scientific programming language Fortran (see `http://www.tutorialspoint.com/fortran/`). Especially this programming language is in the interest of analyses in connection with the implementation of new technologies due to a number of scientific code, occurring in analytical programs, especially in the commercial

sector.

Many other compiled or interpreted programming languages just like C# (see https://msdn.microsoft.com/en-us/library/618ayhy6.aspx), Java (see https://www.java.com/en/about/), Python (see https://www.python.org/) or Matlab
(see http://www.mathworks.com/products/matlab/) etc. contain libraries which support in some form the parallel run of computing tasks. These libraries, especially MPI, are very well known to interested researchers for the parallel run of a variety of computations in computational mechanics, particularly for the domain decomposition method.

Since the year 2007, when the CUDA (Compute Unified Device Architecture, see http://www.nvidia.com/object/cuda_home_new.html) was introduced by NVIDIA company, it has led to significant increases in the possibilities for the utilization of high-performance multi-core graphical chips not only for computer visualization purposes for 2D and 3D computer graphics. CUDA and related technologies like OpenCL (Open Computing Language, free standard for cross-platform, parallel programming of modern processors, see https://www.khronos.org/opencl/, [33]) and Microsoft DirectCompute (supports general-purpose computing on graphics processing units on Microsoft's operating systems since Windows Vista OS, see https://channel9.msdn.com/Tags/directcompute-lecture-series) are currently being investigated by many researchers. Their applications occur in research involved mainly in information technologies, in algorithms for artificial intelligence (diagnosis and synthesis of voice, image recognition, etc.).

Numerical methods which were used in the past due to their inefficiency at the mere periphery of scientific interest, due to the current technological advances have become more attractive. This involves mainly explicit numerical methods used in computational mechanics. Especially it concerns the explicit method used in dynamics, statics of structures and fluid dynamics respectively.

Explicit algoritms are highly suitable for a solution of short time highly non-linear computations mainly for numerical simulation of the processes of forming casts or the simulation of crash tests in the automotive and aviation industry or for shape finding of thin membranes in civil engineering etc. This method facilitates the consideration of a variety of nonlinearities in an easy and explicit manner. Based on the core of explicit dynamic numerical methods is also the dynamic relaxation numerical method which is used for static numerical analysis of civil and mechanical structures. The conditional stability character of explicit methods leads, in some cases, to the necessity to use an exceptionaly

small integration step. As a consequence, explicit methods are time consuming. The availability of a huge number of parallel threads directly implemented in hardware enabled their effective usage (see [104]).

1.2 Motivation

The view is focused on both the various types of hardware and computer networks, resectively. Today's development tools then enable their effective combination and usage. This concerns primarily to the parallelization on graphic processors due to the GPGPU technology and multicore CPUs. Computer networks then allow another level of parallelization of already parallel models, which are applied on the local machine. This is particularly relevant to the issue of so-called the Big Data.

The field of challenging numerical computations of a crash test simulations in the automotive and aerospace industries seems to be a relevant application area. The highly geometrically non-linear behavior of the shell structures of car and aircraft bodies in the respective dynamic processes requires the most effective way of dealing, and parallelism here is more than desirable.

The rapid availability of results from numerical simulations allows more efficient design of the relevant constructions. In the case of transport constructions it is primarily for the purpose of simulation to estimate the indices of impact severity under the conditions of EN 1317 and also other than those. European Norm EN 1317 defines common testing and certification procedures for a road restraint systems. Thus the aims of the thesis can be summarized into two points, as follows:

(i) The main benefit should be to explore the possibilities of computations of parallelism for the use in computations of statics and dynamics of structures within the framework of an explicit numerical methods applied in scope of the Finite Element Analysis (FEA). Based on an analysis of the current hardware and software opportunities, design a parallel approach to enable their efficient utilization.

(ii) Perform numerical simulations of nonlinear dynamics using the explicit form of FEM for appropriate structures with the help of designed software solution.

1.3 Problem Statement

For decades, the vast majority of software applications contained algorithms used for sequential execution of instructions only. Multi-process and asynchronous approaches in programming were generally considered primarily by system engineers for designing operating systems or special applications focused on network resource utilization. To a large extent, however, it was related to a pseudo concurrency (or parallelism) of applications, as the instructions were performed sequentially on a single processor core thread. The impression of concurrency is caused by the rapid alternation of running of the different processes in successive time. For such reason, the division of processor time did not provide any advantage in terms of processing more data, but vice versa. Thus, a large number of software applications for scientific-technical computations have been focused on what they solved rather than how they solved it. Insomuch as it relied heavily on the progressive growth of single CPU performance, there was almost no effort to think about the issues of parallelism and appropriate algorithmization. This was largely due to the purchase price of such computer technology, the limited amount and complexity of software technologies focused on handling such computer systems, and the lack of qualified professionals on the labor market or in academia.

Nowadays, multi-core and multiprocessor computing systems or access to computer networks are an absolute must for everyday life. Therefore, it is an attempt to maximize the usage of such computer performance through applications that are capable of doing so. The reasons are primarily of economic nature, this concerns an increase of a labor productivity and progress in advancement of a developing new technologies for commercial usage.

In conditions of computational mechanics, this concerns mainly analytical software systems using the FEM, finite difference method, finite volume method or their combination to a numerical solution of a set of integral-differential equations often with a nonlinear character. Since many of these software systems have been developed over the decades since the 1960's, thus they contain a large amount of program code with a considerable amount of tuned functionality. The implementation of new approaches related to the parallelization of respective computations requires considerable expert analysis of existing and new algorithms. The same applies to the analysis of suitable hardware and software technologies. Here, it is also necessary to realize the fact, software and hardware technologies are closely interlinked, so the algorithms used must respect all of the technological constraints in order to optimize their effective

usage. The parallel approach has not always been an effective way in effort to improve the performance of affected operations, so the process of searching for opportunities for parallelization is equally important.

At this time, the technologies of multi-core CPU processors, programmable graphical multi-core GPU processors, as well as their combinations, are commonly available on the market. Also due to the affordability of solid computer assemblies, the next level of parallelism is an interconnection of such parallel machines (CPU+GPU) in a generally heterogeneous computer cluster through the computer network. All of the mentioned opportunities and limitations then create huge demands on the complex expert maturity of the responsible person. The situation gets easier if the expertise can be distributed through a team of experts, but it is often not the case, mainly due to the lack of experts on the labor market (situation in the Czech Republic at the time of composing the thesis).

2 Background and State-of-the-Art

2.1 Scientific Computing in Computational Mechanics

During the 60's of the 20th century a variety of programs began to appear, especially in the military field for the US space program and applications, which were focused on a solution to a number of tasks for stress analysis in the mechanics of structures and the dynamics of fluids, as well as automatic control. The American National Aeronautics and Space Administration (NASA) began with the development of the now well-known commercial program NASTRAN (abbreviation of the NAsa STRucture ANalysis, developed by the Computer Sciences Corporation) for its space program, based on FEM, which at that time began to experience a great boom just because of the development of computer technology. The period of the 60's of the 20th century can be considered as one of the most important periods in modern society. Czechoslovakia in this development played a significant role, mainly due to the people working around the Brno University of Technology.

At this time, research institutes in the USA and Europe began to persue the development of FEM and its computer realization. In this regard, the global development centers of FEM belonged the Swansea University (Prof. Olgierd C. Zienkiewicz, ∗1921, †2009) in the United Kingdom, the University of Stuttgart (Prof. John H. Argyris, ∗1913, †2004) in Germany and Brno University

of Technology (Prof. Vladimír Kolář (∗1928, †2000), Prof. Miloš Zlámal (∗1924, †1997), Assoc. Prof. Ivan Němec (∗1945) and Prof. Jiří Kratochvíl (∗1929)) in Czechoslovakia. Later, other prominent scientists joined, mainly researchers from the University of California, Berkeley (Prof. Robert L. Taylor) and MIT (Prof. Klaus-Jürgen Bathe, ∗1943), both from the USA.

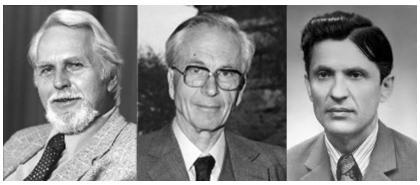


Figure 1: Prof. Olgierd C. Zienkiewicz, Prof. John H. Argyris and Prof. Miloš Zlámal.

As a pioneer in the development of the world's first FEM programs was Prof. Edward L. Wilson (∗1931) from University of California, Berkeley. His first programs had no name, later they were transformed into the system under the name SAP (abbreviation of the Structural Analysis Program) and NONSAP (abbreviation of the NONlinear Structural Analysis Program). As the first commercial FEM software for solving nonlinear problems can be considered program MARC developed by the Marc Analysis Research Corporation named after its creator Pedro V. Marcal from Brown University.

With regard to the explicit computational form of the FEM, it is necessary to remember the program SADCAT, which was developed in the early 1970s at Argonne as one of the first three-dimensional structural dynamics programs which employed explicit time integration to achieve great efficiency for the computational simulation of impulsively loaded shells. This program closely relates to the work of Prof. Ted Bohdan Belytschko (∗1943, †2014) from the Northwestern University, who focused on the development of effective finite elements for transient nonlinear dynamics. It is important to note his effective C^0 triangular shell finite element, which is used in numerical simulations in this thesis (see [136]).

For completeness, it is necessary to mention the well-known programs based on FEM, which at that time and some years later began to emerge. It is a group of programs e.g. ABAQUS, ANSYS (founded by John A. Swanson), ADINA see `http://www.adina.com/index.shtml` (founded by Prof. Klaus-Jürgen Bathe), and especially software DYTRAN for explicit dynamics and fluid struc-

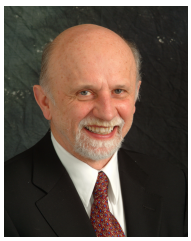


Figure 2: Professor Ted Bohdan Belytschko.

ture interaction (see `http://www.mscsoftware.com/product/dytran`).

From a national perspective, it is necessary to acknowledge the work of Prof. Vladimír Kolář and Assoc. Prof. Ivan Němec, who began with the development of software tools for a solution to civil and mechanical structures by FEM.

The NE-XX system (NE means initial letters of the author's surname Němec, see `http://www.fem.cz/historie/?lang=en`) became a widespread computational analytical tool based on FEM, both in Europe and now around the world. The NE-XX system is now the numerical computational core of the commercial program RFEM (see `https://www.dlubal.com/en/rfem-5xx.aspx`) of the German company Dlubal Software and the Scia Engineer (see `http://www.scia.net/en/software/product-selection/scia-engineer`) of the Belgian company Nemetschek Scia which occupy an important position in the European marketplace with this type of software.

Professor Vladimír Kolář belongs among the pioneers of FEM. He was persecuted and removed from academic life during the totalitarian communist regime after the Soviet invasion in the year 1968. The publication of his books and articles were prohibited.

Despite all the hardships he had educated many of their successors through the secret lectures. He founded the famous Brno school of FEM which has gone on to become top class. In the time of freedom, his name, legacy and glory live on again, without persecution of the totalitarian regime. A team of academicians from the Brno University of Technology, which was built around his person, significantly contributed to the theoretical and a practical development of FEM in a global context.

In connection with Prof. Vladimír Kolář acknowledgement also need to be given to the work of Prof. Ivo Babuška (∗1926) from the University of Texas, Austin and Prof. Jindřich Nečas (∗1929, †2002) from the Faculty of

Figure 3: Professor Vladimír Kolář in 1976.

Mathematics and Physics of Charles University.

It is also necessary to mention the development of software tools at the Czech Technical University in Prague (CTU in Prague) in the Department of Mechanics, Faculty of Civil Engineering. These are primarily general analytical software tools OOFEM (abbreviation of the Object Oriented Finite Element solver, see `http://www.oofem.org`, [98]) developed by Prof. Bořek Patzák and SIFEL (abbreviation of the SImple Finite ELements, see `http://mech.fsv.cvut.cz/~sifel/`) developed by Prof. Jaroslav Kruis. Especially, it is necessary to mention the work of Prof. Jaroslav Kruis focused on the parallelization using the FETI based method (abbreviation of the Finite Element Tearing and Interconnecting, [63]). Also worth mentioning is the project MuPIF (abbreviation of the Multi-Physics Integration Framework) for implementation of multi-physics and multi-level simulations assembled from independently developed applications components (see [99]).

Among the international open source projects it is necessary to mention the project Kratos (see [68], [53] atc., `http://www.cimne.com/kratos/`), which is a framework for building multi-disciplinary finite element programs. It is free multi-physic FEM C++ open source code. Kratos is parallelized for Shared Memory Machines (SMMs) and Distributed Memory Machines (DMMs).

Kratos is being developed within the framework of the international project CIMNE (The International Center for Numerical Methods in Engineering, research organization created in 1987 at the heart of the prestigious Technical University of Catalonia (UPC) and a Partnership Between the Government of Catalonia and the UPC; see `www.cimne.com`). The aim of CIMNE is the development of numerical methods and computational techniques for advancing

knowledge and technology in engineering and applied sciences.

From the point of view of the utilization of the GPGPU for numerical computations, it is necessary to mention the open-source finite element toolkit NiftySim (see `https://sourceforge.net/projects/niftysim/`). The toolkit is founded on the total Lagrangian explicit dynamics (TLEDs) algorithm. Total Lagrangian explicit dynamics is an efficient and accurate approach for simulation of soft tissues in biomedical applications (see [56]) as well as for the contact/impact simulations of various structures. Another open-source finite element toolkit based on explicit dynamics is project Impact (see `http://www.impact-fem.org/`).

2.2 Multiprocessor and Multicore Technologies in Scientific Programming

At first, it is necessary to mention supercomputers based on the multiprocessor architecture, and currently even in combination with the multicore processor technology. These include supercomputer ILLIAC IV as a first massively parallel computer (University of Illinois, [16]); The UNIVAC division of Sperry Rand Corporation delivered the first multiprocessor 1108 containing up to 3 CPUs and EXEC 8 OS supporting multithread program execution for respective hardware [38]; PEPE (shortcut of the Parallel Element Processing Ensemble) developed for the ballistic missile defense environment [22]; The Connection Machine [44] as a series of supercomputers that grew out of Danny Hillis's doctoral research at MIT in the early 1980s; BBN Butterfly built by Bolt, Beranek and Newman [74] in the late 1980s; Evans and Sutherland who came up with the world's first general supercomputer machine (ES-1) on market in 1989 [117]; Intel iPSC (Intel Personal SuperComputer) [25]; etc.

Currently, the most powerful computers belong to Titan-Cray XK7 (built by Cray at Oak Ridge National Laboratory in the USA), which is designed for various physical simulations (modeling global atmospheric phenomena, nuclear physics, etc.) and the most powerful supercomputer Tianhe-2 (MilkyWay-2) in China.

On the other side stands the technology of quantum computers [67] in comparison to the already mentioned computer technology. Respective technology is built principally on a different basis than the current digital computers. The first such commercially used quantum computer is D-Wave of the world's first quantum computing company D-Wave Systems in Canada. Their systems are being used by world-class organizations and institutions including Lockheed-Martin

Figure 4: The world's fastest and most powerful digital supercomputer Tianhe - 2.

(first customer), Google, NASA, and USRA (abbreviation of the Universities Space Research Association).

Quantum computing is absolutely a breakthrough technology that is capable of performing amazing computations, unthinkable with current digital computers. The computer D-Wave is currently not a device for a general-purpose application as is the case of supercomputers Tianhe-2 and the others. Due to the high complexity of such devices, they are designed to solve specific optimization problems where they effectively use the main idea of quantum theory.

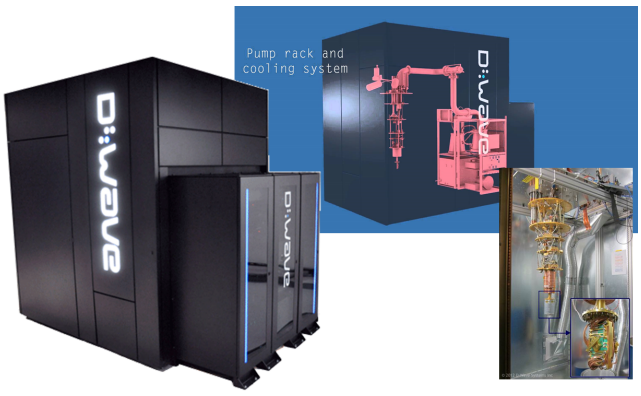


Figure 5: Quantum computer D-Wave.

In contrast with the above, the most advanced technologies for parallel data processing are personal computers (PC) that had been built for many years on a single core CPU and single processor architecture. Thus a large part of the development of the FE tools had been primarily concentrated on the sequential code execution. Multiprocessor systems have existed for decades, but until recently they were mostly found only in supercomputers and in large server systems. Multicore desktop computers, and even multicore embedded devices, are now increasingly prevalent. Until the advent of multicore processors in the first decade after the year 2000, it was not possible to use the real parallelism on common PCs.

In the previous years, many algorithms had begun to be developed which allowed parallel processing of computational tasks. These are primarily domain-decomposition methods, which mainly relate to graph partitioning of the finite element mesh (see below [59], [11], [126]). An important numerical method related to a domain decomposition, it is the FETI method for the numerical solution of a large linear systems arising in linearized engineering problems, which in the early 90s developed C. Farhat and F.X. Roux [29]. The FETI method was originally proposed as a dual discrete nonoverlapping domain decomposition method for the parallel finite element solution of static equilibrium equations. Later arose its various modifications as TFETI (abbreviation of the Total FETI; see [23]) or HTFETI (abbreviation of the Hybrid Total FETI; see [110]). These are the computations performed solely on the CPU (MPI, OpenMP, etc.). Later, the FETI method was used for the other purposes than only for a parallel solution of a finite element models. These are primarily the work of scientists from CTU in Prague, namely the teams around Prof. Jaroslav Kruis and Prof. Zdeněk Bittnar (see [63], [64], [37]).

With the advent of technologies supporting general computation on graphic cards (GPGPU), the development of algorithms has began for image processing (see [51], [100]), algorithms for numerical mathematics and linear algebra (see [76], [96], [88] [2], [19]). Another field of application are numerical solvers for partial differential equations (see [26], [128], [119]). A high degree of importance is put on software applications for CFD, where the demanding requirements for the sparsity of the finite volume of finite element mesh and thus the high requirements for numerical processing of the problem (see [127], [75], [55], [111], [101]). Another field of application relates to the particle dynamics in many branches of physics (see [78], [73], [107]). Last but not least, inventions in the field of a code transfer between various software technologies and their combinations, is needed to be cited (see [24], [35]).

3 Overview of Our Approach

3.1 Mathematical-Physical Model

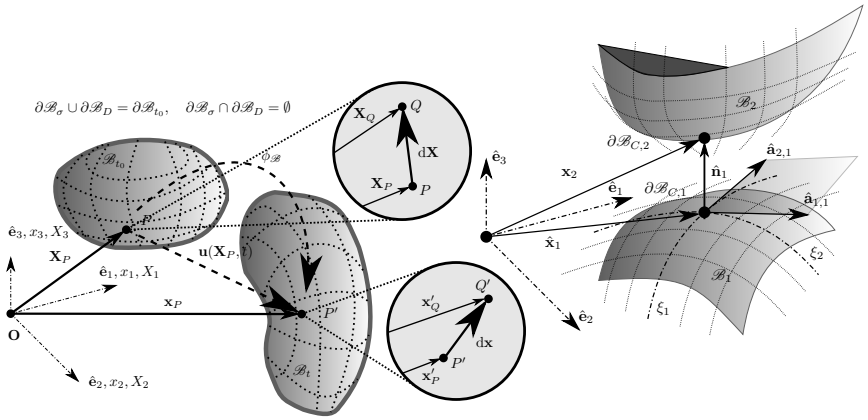


Figure 6: The solid body kinematics.

The most general formulation of problems in continuum mechanics leads in the case of the dynamics of deformable bodies to the shared formulation with the dynamics of fluids, which is generally known by the name of Navier-Stokes equations. Equations can be expressed in differential form (Eulerian space description), which are called *Cauchy's equations of motion*

$$\boxed{\operatorname{div} \boldsymbol{\sigma}^T + \varrho\mathbf{b} = \varrho\mathbf{a} \text{ in } \mathscr{B} \times [0,\tau]}.$$

For the solvability of these equations it is necessary to prescribe the essential and natural boundary conditions and initial condition at time t_0

$$\mathbf{u}(\mathbf{x},t) = \overline{\mathbf{u}} \text{ on } \partial\mathscr{B}_D \times [0,\tau], \ \mathbf{t} = \boldsymbol{\sigma}\cdot\hat{\mathbf{n}} \text{ on } \partial\mathscr{B}_\sigma \times [0,\tau],$$
$$\mathbf{v}(t_0) = \overline{\mathbf{v}}_0 \text{ in } \mathscr{B}, \ \mathbf{u}(t_0) = \overline{\mathbf{u}}_0 \text{ in } \mathscr{B}.$$

The problem being studied generally contains various possible types of nonlinear behavior. Solving such problems requires advanced tools of mathematical analysis. This analysis is necessary especially with regard to the numerical solution of IBVP (abbreviation of the Initial Boundary Value Problem). These are

mainly the finite element method, which is directly related to the issue of the relationship between a strong solution in terms of a general analytical solution of a partial differential equations and their solution by a so-called weak formulation. In mathematical physics, this type of analysis relates mainly to finding solutions to some of the functional as its stationary point.

Theorem. *Let* $\mathbf{q}(t) : [t_0, t_1] \to \mathbb{R}^n$ *be a function of class* $C^2[a, b]$ *satisfying the boundary conditions* $\mathbf{q}(t_0) = \overline{\mathbf{q}}_1, \mathbf{q}(t_1) = \overline{\mathbf{q}}_1$. *Then the first variation of the functional* $\mathfrak{J}(\mathbf{q}, t)$ *is given by the functional gradient, and leads to the form of Euler-Lagrange equations.*

$$\boxed{\delta\mathfrak{J}(\mathbf{q}(t), t) \equiv \partial_{\mathbf{q}}\boldsymbol{\mathcal{L}}(\mathbf{q}(t), \partial_t\mathbf{q}(t), t) - \frac{\mathrm{d}}{\mathrm{d}t}\left(\partial_{\partial_t\mathbf{q}}\boldsymbol{\mathcal{L}}(\mathbf{q}(t), \partial_t\mathbf{q}(t), t)\right)},$$

where $\boldsymbol{\mathcal{L}}(\mathbf{q}(t), \partial_t\mathbf{q}(t), t) \in C^2[a, b]$ is the associated Lagrangian.

Hamilton's variational principle as a leading variational principle in the thesis is used. It is deduced from an analogy to the Lagrangian form of D'Alembert's principle. The analogy in elastodynamics corresponds to the Bubnov Galerkin weighted residual form, which leads to the principle of virtual work. The weak formulation of Cauchy's equation of motion has the following form

$$\delta\int_{t_0}^{t_1}\int_{\mathscr{B}}\left(\operatorname{div}\boldsymbol{\sigma}^T + \varrho\mathbf{b} - \varrho\mathbf{a}\right)\,\mathrm{d}v\,\mathrm{d}t = 0,$$

$$\delta\int_{t_0}^{t_1}\boldsymbol{\mathcal{L}}\,\mathrm{d}t + \delta\int_{t_0}^{t_1}\mathcal{W}\,\mathrm{d}t = 0, \quad \delta\mathbf{u}(t_0) = \delta\mathbf{u}(t_1) = 0,$$

where

$$\boldsymbol{\mathcal{L}} = \underbrace{\frac{1}{2}\int_{\mathscr{B}}\varrho\mathbf{v}\cdot\mathbf{v}\,\mathrm{d}v}_{\mathcal{K}} - \underbrace{\int_{\mathscr{B}}\varrho\Psi\,\mathrm{d}v - \mathcal{U}_c^P}_{\mathcal{U}}, \ \mathcal{W} = \int_{\mathscr{B}}\varrho\mathbf{b}\,\mathrm{d}v,$$

where $\mathcal{K}$, $\mathcal{U}$ and $\mathcal{W}$ denote kinetic energy, deformation energy with energy of contact forces and energy of applied external forces respectively. The term $\varrho\Psi$ denotes density of the Helmholtz free energy and $\mathcal{U}_c^P$ denotes energy potential of the contact forces which are considered in the form of Hertz–Signorini–Moreau (or Kuhn–Tucker–Karush) conditions which have the following form

$$g_N \geq 0, \quad p_N \leq 0, \quad p_N g_N = 0$$

$$g_N = \begin{cases} (\mathbf{x}_2 - \hat{\mathbf{x}}_1) \cdot \hat{\mathbf{n}}_1 & \text{if } (\mathbf{x}_2 - \hat{\mathbf{x}}_1) \cdot \hat{\mathbf{n}}_1 < 0 \\ 0 & \text{otherwise.} \end{cases}$$

$$\delta \mathcal{U}_c^P = \delta \left(\frac{1}{2} \int_{\partial \mathscr{B}_c} \epsilon_N g_N^2 \, \mathrm{d}a \right) = \int_{\partial \mathscr{B}_c} \epsilon_N g_N \delta g_N \, \mathrm{d}a$$

Contact conditions are critical area in dynamic simulations under the consideration. The contact causes the formation of additional forces, which at the time of their creation are suitably applied into the process of numerical integration of the equations of motion. Here, the problem of general contact is considered, especially for the purposes of collision with the barrier and subsequent post-critical behavior of the respective structure.

3.2 Numerics of Applied Mathematical-Physical Model

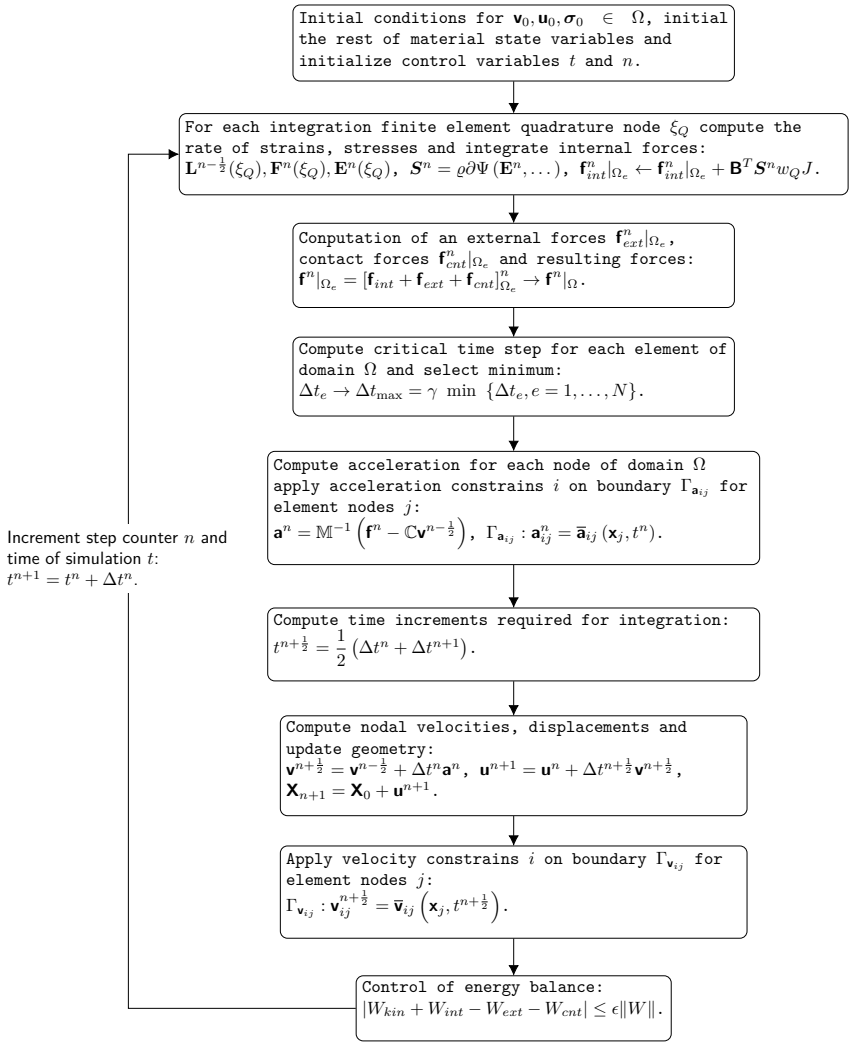


Figure 7: Flowchart of an explicit time integration algorithm.

For the purpose of designing and testing a hybrid-parallel computing model of an explicit numerical solver related to the problem solution of nonlinear dynamics using FEM, an effective triangular shell finite element developed and published by T. Belytschko, H. Stolarski and N. Carpenter in 1984 (see [8]) as complement to the Belytskchko-Lin-Tsay quadrilateral shell element (see [6]) is applied with corotational coordinates for *small strain* and *large rotation* kinematics (see [58], [5]).

Cauchy's equations of motion represent a dynamic equilibrium in the current configuration of a body $\mathscr{B}_t$. Respective body $\mathscr{B}_t$ is represented by the domain of its finite element discretization Ω. It also applies even for body boundaries $\partial\mathscr{B}$ which are represented by Γ. After application of FE discretization to respective functionals it leads to the set of second order semidiscrete nonlinear ordinary differential equations (ODE) later used for direct integration by the *central difference method*. If the given ODE are nonlinear, then the resulting form is thus the following

$$\mathbb{M}\ddot{\mathbf{u}} + \mathbf{f}_{int}(\mathbf{u}) = \mathbf{f}_{ext}(\mathbf{u}) \in \Omega.$$

The central difference method with variable time step as a popular method in computational mechanics is considered. It is used for discretization velocities and accelerations in the terms of Lagrangian meshes (see [7], [83], [39]). In respective time integration algorithm it is necessary to control stable time step as the mesh deforms and the wave speed changes due to the stress. Respective algorithm is shown in Fig. 7, where the term $\mathbb{C}\mathbf{v}^{n-\frac{1}{2}}$ represents additional damping forces explicitly included in equations of motion.

3.3 The Main Used Algorithms

3.3.1 Searching in the Euclidean Space

For the purpose of finding a solution to many problems within the scope of computer science, the respective problem can often be transformed into the task of searching in an appropriate mathematical space. This is also the case in Computational Contact Mechanics (CCM). Due to the considerated generality of a model of contact, the algorithm used must be applied to all the FE nodes of the FE model, which is the most computationally demanding process. More than any of the areas of CM (abbreviation of the Computational Mechanics), contact problem is the most closely related to the field of theoretical computer science through the branch of computational geometry.

The solution to the problem of contact of solids then theoretically falls within the area of a so-called *nearest neighbor* (NN) search. The core of such an algorithm is here defined as a collection of n objects (FE nodes) that build a data structure, which provides objects (FE nodes, FEs, etc.) in the time as fast as possible based on the NN *query* represented by the FE *bounding box*.

Compared with older algorithms for a solution of contact problem, the proposed algorithm uses topology mapping based on the *kd*-tree data structure applied in database software systems or graphical engines for quick processing of the range-searching query over the d-dimensional data. This type of data structure represents special case of binary trees. Thus it provides effective logarithmic type of time complexity for search problem.

Subsequent post-processing represented by the computation and application of induced contact forces within the explicit integration algorithm meets the standard approach usually applied in a number of commercial software packages for the numerical simulation of crash tests by FEA in the automotive and aviation industries, and also in traffic structure engineering for the design of a road restraint systems.

Respective algorithm is also critical for analysis of *MEIM* (abbreviation of the Macro Entity Interaction Multigraph), which is one of the cornerstone steps in hybrid-parallel type of computation developed.

3.3.2 Analysis of the Macro Entity Interaction Multigraph

The proposed algorithm for the purpose of hybrid-parallel computational processing strongly relates to the analysis of the motion the individual macro elements in a space representing the structures interacting with each other through the contact forces.

Nonlinear dynamics is often externally demonstrated by the chaotic behavior providing the so-called *strange attractors*. As the source of chaotic behaviour, is here, provided by mutually interacting deformable bodies, which generate special types of combinatorial sequences. Such a combinatorial sequence can be defined by so-called *unoriented multi-graph*. The topology of such a multigraph can be defined by the spatial distribution of the interacting bodies. The nodes in the multi-graph represent all the contained bodies (macro entities) and the graph edges represent contact interaction between the bodies.

The MEIM assembly is performed through the range searching queries to the kd-tree data structure used for mapping the spatial data. The MEIM analysis itself is further based on the *depth-first search* (DFS) algorithm to find *connected*

subgraphs representing individual clusters of touching bodies.

The assembly and analysis of the MEIM is only required while performing a numerical simulation in the scope of a computer network where it is no longer possible to take advantage of the shared memory address space. The algorithm for the data distribution within a computer network starts primarily from the basic DFS analysis of the MEIM. Based on the DFS algorithm, all connected subgraphs are obtained, even if they represent isolated nodes only.

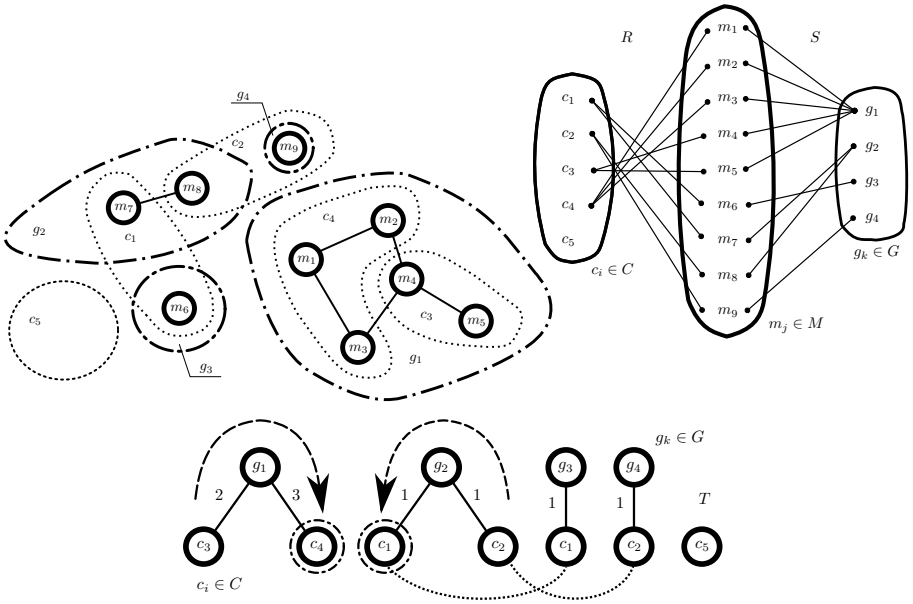


Figure 8: The MEIM model example, where $c_i \in C$ (workstations connected to computer network), $m_j \in M$ (macro entities) and $g_k \in G$ (connected subgraphs).

4 Main Results

4.1 Parallel Hybrid Model

Since the implementation of the discussed algorithms and technologies in existing open-source projects mentioned earlier would require considerable time for their own analysis, a hybrid-parallel numerical testing solver called FEXP (abbreviation of the Finite [E]lement [E]XPlicit solver) was composed. The solver has a testing purpose, which means testing the effectiveness of both the algorithms and the technologies used.

The FEXP solver for parallelization of computations combines the possibility of using multi-core processors (CPU) with the possibility of parallelization in computer heterogeneous cluster interconnected in a LAN. In the case of a local workstation within the LAN, it concerns the use of multicore CPUs and the later possible usage of GPGPU technology Nvidia CUDA or OpenCL refering to the use of Graphics Processing Units (GPU) for some parts of the computations.

With respect to generality, availability and support of a wide range of software technologies and software modularity, the programming language C++ in current version 14 was chosen as the main programming language for the composition of FEXP solver. As a development environment, the Windows OS was chosen, primarily due to the availability of advanced development tools required to develop such applications. Thus for the project management and code compilation, IDE Microsoft Visual Studio 2015/2017 Community was used. For the purpose of easier user control over the FEXP solver, the windows based application FEXP Solver Manager was designed and programmed using Microsoft .NET/C# programming language requiring .NET Framework 4.6 (C# 6.0) or later versions.

For the mentioned reasons, the FEXP solver is divided into two main parts from an external viewpoint. To the parts refering to the functionality running on a single workstation and the part of the networking extension. This model is illustratively shown in the Fig. 9.

From the Fig. 9, it is noticeable that the major part of all computations is done on the single workstation. The server computer then works primarily as a final assembler and distributor of a overall work. From this perspective, the FEXP model is comparable to the massive parallel processing model used in the LS-DYNA under the ANSYS LS-DYNA Parallel license (see `https://www.ansys.com/products/platform/ansys-high-performance-computing`). However, this approach requires third-party MPI software

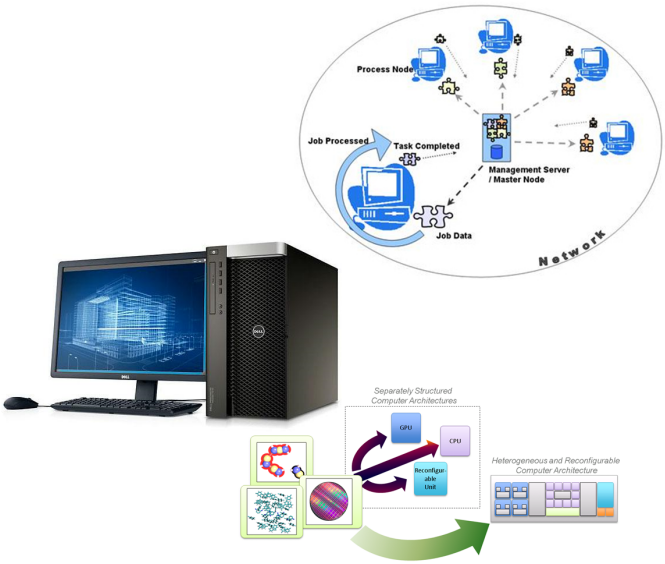


Figure 9: Simplified representation of designed parallel model of the FEXP solver.

support be correctly installed (see `https://www.ibm.com/us-en/marketplace/spectrum-mpi` or `https://software.intel.com/en-us/articles/intel-mpi-library-documentation/`).

The parallel model of FEXP is designed somewhat more generally in this case, initially with greater emphasis on efficient utilization for explicit impact dynamics. Unlike LS-DYNA, it does not have to use the demanding methods of domain decomposition. The critical part of the LS-DYNA parallel model is based on a domain decomposition (decomposition of meshes) using METIS (Serial Graph Partitioning and Fill-reducing Matrix Ordering, see `http://glaros.dtc.umn.edu/gkhome/metis/metis/overview`) similarly as a free open-source finite element program SIFEL that uses the open source library HPMESHDECOMP based on METIS (see `http://mech.fsv.cvut.cz/~sifel/TOOLS/meshdecomp.html`). In the case of program SIFEL, the FETI method is then applied on decomposed model.

The parallel FEXP model is based on a physically logical decomposition

according to the specific nature of the addressed impact problem. This approach is applicable primarily to the simulation of the impact tasks of a wider group of separate entities interacting with each other through contact forces. As an example in this context, collision of many vehicles on the highway could be mentioned. Throughout the addresed models, the critical part of an effective contact detection must be considered. This of course also applies to the so-called *self-contact* problem. For such a reason, an algorithm for the so-called *nearest neighbor searching* is applied.

Since the parallel hybrid model is based on the interaction of distinct entities, so for the purpose of computation on the computer network, it was necessary to design a heuristic approach for computer network balancing based on an analysis of the *macro entity interaction multigraph* (MEIM). An analysis of the MEIM provides a scope for future optimization, both in terms of the quantity and character of the data being considered, as well as different types of heuristics such as their popular subset of the so-called *metaheuristics*, often used in *soft computing*. The applied algorithm for the MEIM analysis will then be further presented in subsequent chapters as well as the algorithm for self-contact and also for macro body contact detection.

4.2 Description of the FEXP Parts

The FEXP solver architecture is presented in the Fig. 10 in its simplified form. It schematically describes main thematic blocs contained in the program. As shown the FEXP solver is modularly composed from the separate parts meeting the required functionality explicitly defined by a specific interface for each such part. For better understanding of the FEXP solver software architecture, a rough explanation of an individual parts then follows.

- *Common and Concurency* libraries include both the functionality generally used throughout the FEXP solver, as well as functionality related to the work with threads and asynchronous operations. It also includes resources for safe management with the dynamically allocated memory, it relates especially to the usage of so-called *smart pointers*.
- *File handling, Input and output data* libraries includes both the functionality focused on handling with input and output files (preprocessing and post processing) and important part related to data tables assembly.
- *Network communication, Network server, Analysis of MEIM* libraries include both inter-process communication functionality for the scope of the

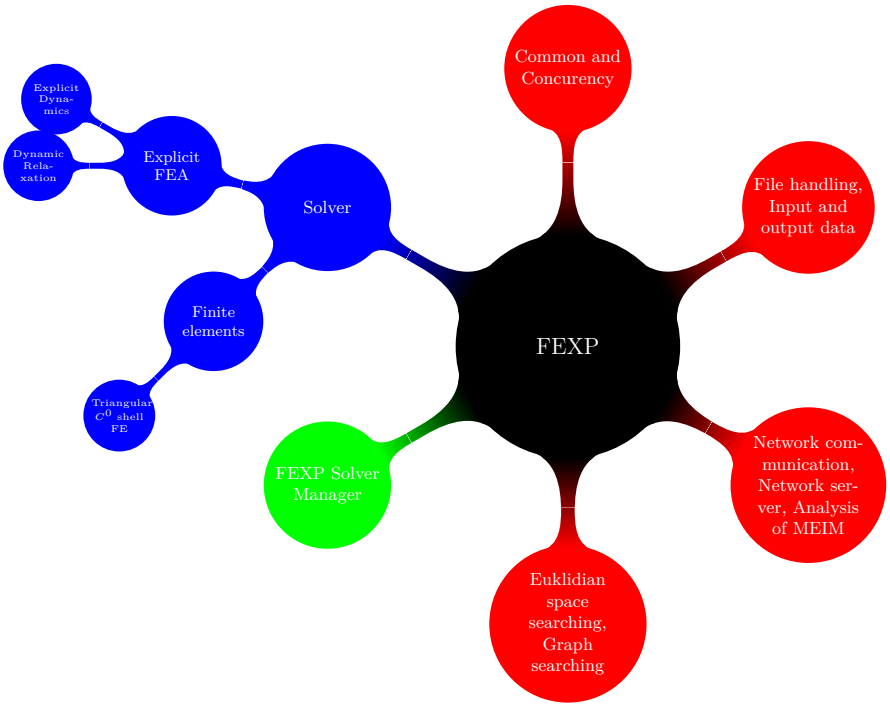


Figure 10: Composition of the FEXP solver.

computer network and the analysis-related functionality required for the data exchange within the considered computer network. It is based on the analysis of a non-oriented graph representing the interaction of individual bodies (MEIM) during the numerical simulation. Here, it is necessary to also include the implementation of the network server.

- *Euklidian space searching, Graph searching* libraries include both the basic algorithms required for the general solution of the nonlinear contact problem, as well as algorithms required for the representation of the non-oriented interaction multigraph and the basic algorithms for their analysis.

- *Solver* includes the functionality required for the numerical simulation

of dynamic processes. It concerns in particular the library containing functionality for the integration of specific FEs, as well as the functionality that controls the process of explicit numerical integration of equations of motion in terms of the FEA. It includes the parts designed purely for the calculation in a single workstation, as well as the part concerning the computations within a computer cluster.

- *FEXP Solver Manager* is an external graphical UI based application written in the C# programming language. It is not a necessary part of the FEXP solver solution. It is intended mainly for simplifying the work with the FEXP solver and for presentation purposes.

The listing above contains a basic view of the FEXP solver software solution. It is only a simplified look into the entire designed software solution, especially due to its huge overall size. However, despite considerable simplification, it is possible to subsequently identify and understand the individual parts of the source code.

The following text of this chapter is focused primarily on those parts that are important to understand the entire FEXP software solution. The first part deals with the definition of the type and the character of the input data, the next part relates to the numerical simulation itself, and the final part deals with the presentation of the results. The numerical simulation section is then described in a more detailed fashion, all with regard to parallel data processing. A significant part is then devoted primarily to the hybrid-parallel form of the FEXP solver. It deals with numerical computations within the heterogeneous computer cluster. The FEXP Solver Manager section is appropriately included throughout the text as suggests its purpose.

Where appropriate, the source code fragments are included in each section to give better insight into the implementation of the particular portion involved. With respect to the considerable amount of a source code lines. An effort is made to highlight only those parts of the source code representing an important point for the programmer's orientation. The relational UML diagram is then presented at the beginning of given sections to present the object relationships of an appropriate program section. For the purpose of assembly UML diagrams, the program tool Visual Paradigm (`https://www.visual-paradigm.com/`) was used. The given program is a commonly used tool for object oriented design purposes while software solution designing process.

4.3 Results of Simulation Test

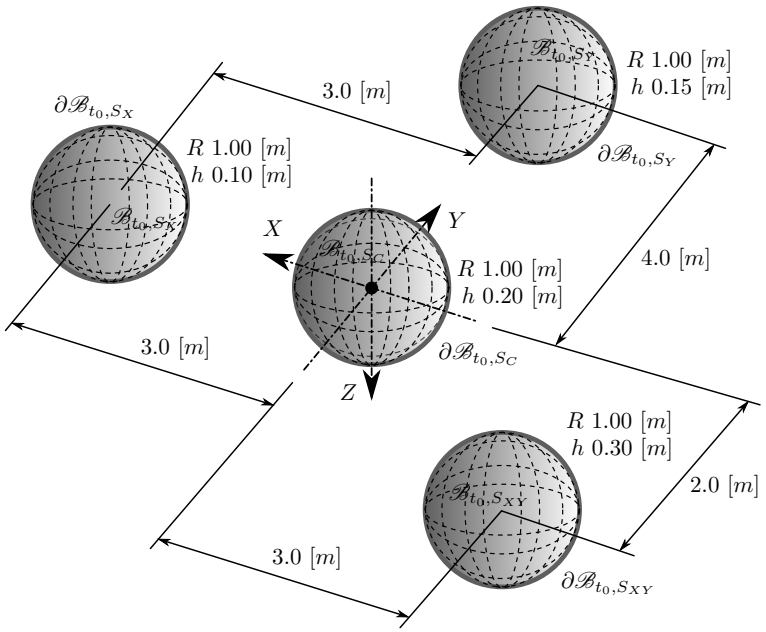


Figure 11: Configuration of spheres at time t_0.

For the testing purposes of the proposed solution represented by the hybrid-parallel FEXP solver, the problem of multibody impact was chosen. The geometrical data of the respective bodies of the mathematical-physical model and their initial configurations are shown in the Fig. 11. This type of example was chosen primarily due to the expected fluctuation of the individual parts of the model within the simulated computer network during the solution process.

The model geometry and FE meshes were created in the RFEM program. The triangular FE meshes were then extracted from files [*.XYZ] (coordinates of FE nodes) and [*.E2D] (connectivity of 2D FE), respectively. The respective files are used by the numerical kernel (the NE-XX solver from the FEM consulting company) of the RFEM program for the FE model assembly process. The FE statistics are included in the Tab. 1.

Table 1: FE mesh statistics.

	Statistics
2D finite elements	768 (≈ 0.016 $[m^2 m^{-2}]$)
FE mesh nodes	392

The movement of bodies is initiated by the initial conditions represented by the velocity constraints shown in the Fig. 12, where small velocities in Z direction are introduced primarily due to the applied type of contact detection algorithm which is represented by node-to-element contact. This artificial numerical impurity avoids the state represented by element-to-element contact.

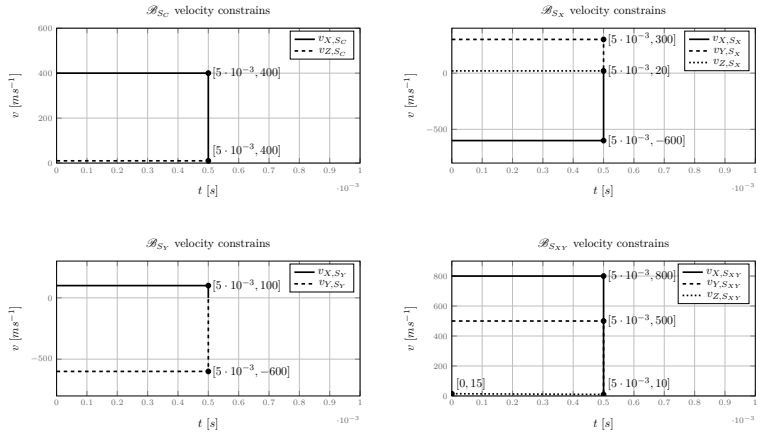


Figure 12: Velocity initial conditions of individual macro entities.

All velocity constraints are switched off at the time $t = 0.0005s$, and subsequently, all of the bodies can move freely based on their inertia. The high initial body velocities are chosen primarily due to the short duration of the transient dynamic simulation and due to their higher destructive effect. For all macro entities (bodies) the same type of material is chosen. The material characteristics are listed in the Tab. 2.

Table 2: Material characteristics.

Characteristics	Value	Unit
E	210.0	GPa
G	80.8	GPa
ν	0.3	$-$
ϱ	7850.0	$kg\ m^{-3}$

For network type computations, a computer grid consisting of 3 workstations is chosen for the initiation of the computation. The 4th workstation is connected to the grid during the computation. The initial configuration of the given computer network is schematically shown in the Fig. 13. It shows the initial placement of the macro entity data over the network.

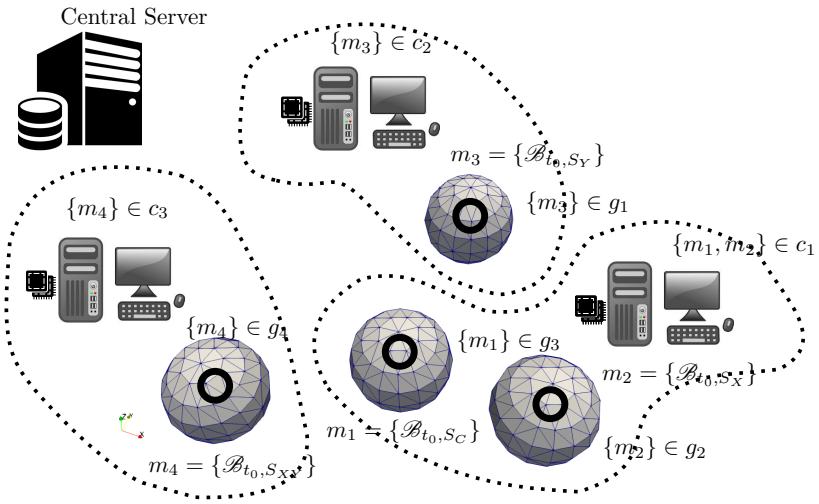


Figure 13: Macro entity distribution over the network at time t_0.

Simultaneously with the parallel processing of tha data over the computer network, the calculation takes place in parallel even within each individual workstation. The number of threads is set uniformly to 3 parallel running threads,

i.e. 2 threads for computation and 1 thread for communication with the server. During the calculation, another client workstation was connected to the server. This workstation acquired the necessary data (so-called default data) to be ready for its eventual involvement into the computation.

The model data distribution over the network represented by the initial state lasted until the time step t_{51}, when the first data transfer was initiated based on the analysis of MEIM. At this time step, the computational data of the m_3 model from workstation c_2 was transferred to workstation c_1. However, the new data distribution over the network lasted for only a short time, and at the time step t_{53} another transfer of model data was initiated again. At this time step, the computational data of the m_4 model from workstation c_3 was transferred to workstation c_1. The two previously mentioned data transfers together with the subsequent new data distribution over the network are represented by the Fig. 14.

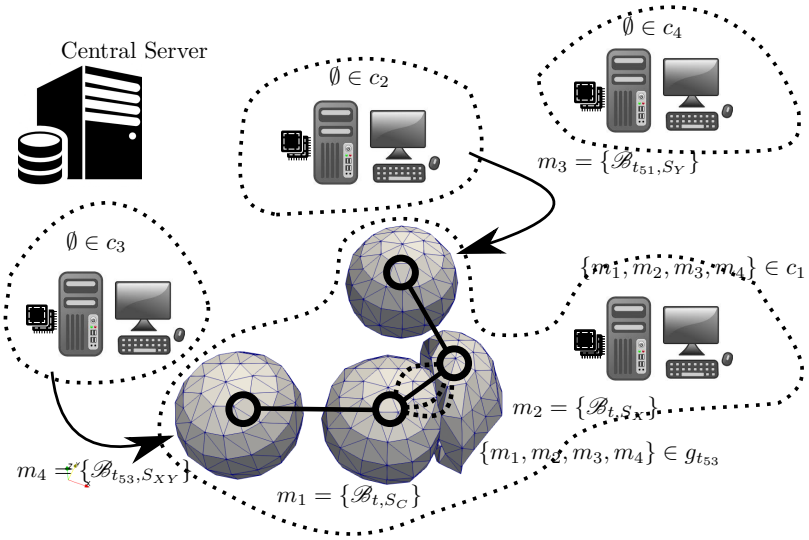


Figure 14: Macro entity distribution over the network at times t_{51} and t_{53}.

After the data transfer from time t_{53} a longer period of time follows in which there is a massive mutual contact interaction between all macro entities. During this time when only one workstation is loaded by numerical computation, all

other workstations perform necessary synchronization procedures only.

During the mutual contact interaction of macro entities both a change in their momentum and direction of movement was caused. Thanks to these changes in motion, individual macro entities began to move apart. At time t_{337}, based on the analysis of MEIM, it was assessed that further data transfer is required. In this case, however, it was the most massive data transfer over the network. It was decided that it is necessary to transfer the data of macro entities m_1, $m2$ and m_3 from the workstation c_1 to the workstations c_2, c_3 and c_4, respectively. This event, along with the new data distribution over the network, is presented in the Fig. 15.

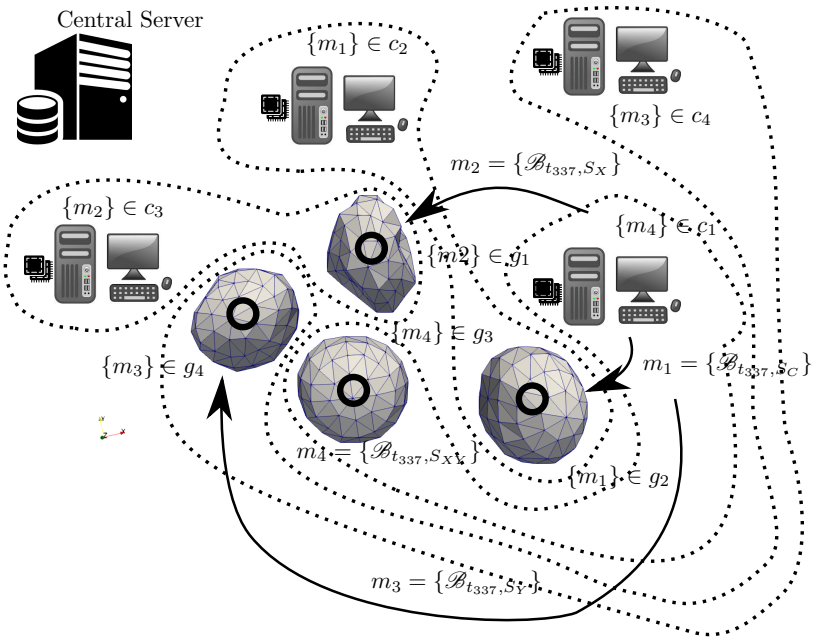


Figure 15: Macro entity distribution over the network at time t_{337}.

The newly acquired state at time t_{337} then lasted until the end of the numerical simulation, i.e. until the time t_{429}. In conjunction with the given model it can be stated that $\approx 66\%$ of the total time of the numerical simulation was spent on one workstation (c_1), i.e. in the time interval from t_{53} to t_{337}. It

may be naive to say that about $\approx 34\%$ of the entire computational time has been saved compared to the numerical simulation, which would take place on one workstation only. Here, of course, it is necessary to include the time spent relating to the data transfers between workstations and the server. However, it can be concluded that in the case of larger models, lower number of data transfers and with the better heuristics related to the analysis of MEIM, it is possible to significantly accelerate the time spent on numerical computations of a similar type of problem.

The input data files for the numerical computation can be found in the attachement of the thesis. This also applies to the result files for the visualization of the respective dynamic simulation in the Paraview program. In this context, videos containing the respective dynamic simulation were created. They are also part of the attachment. Fluctuation of model's macro entities within the computer network during the numerical solution process represents the Tab. 3.

Table 3: Model's macro entities fluctuation within the computer network (state after transfer of macro entities).

Time	**Workstation**	**Workstations content**
t_0	c_1	$\{m_2\} \in g_2, \{m_1\} \in g_3$
	c_2	$\{m_3\} \in g_1$
	c_3	$\{m_4\} \in g_4$
t_{51}	c_1	$\{m_1, m_2, m_3\} \in g_1$
	c_2	$\emptyset$
	c_3	$\{m_4\} \in g_2$
	c_4	$\emptyset$
t_{53}	c_1	$\{m_1, m_2, m_3, m_4\} \in g_1$
	c_2	$\emptyset$
	c_3	$\emptyset$
	c_4	$\emptyset$
t_{337}	c_1	$\{m_4\} \in g_3$
	c_2	$\{m_1\} \in g_2$
	c_3	$\{m_2\} \in g_1$
	c_4	$\{m_3\} \in g_4$

5 Conclusions & Future Work

5.1 Summary

In the dissertation thesis the current issues of the utilization of parallel computations for the purposes of numerical simulations of dynamic processes for assessment, design and optimization of structures in civil and machinery engineering, respectively, are investigated. In view of the current security requirements concerning e.g. the behavior of road restraint systems in the event of accidental vehicle collisions, the safety of vehicles regarding both pedestrian safety and the safety of passengers inside a vehicle in the event of a crash, the safety of strategically important civil structures (nuclear power plants, water supply-related strategic structures, etc.) against the deliberate criminal attempt for destruction related mainly to the problem of world terrorism, thus the numerical simulation of a contact/impact problems became the main interest. Considered type of a numerical algorithm used for the simulation of contact/impact physical phenomana can also be used for rapid dynamic simulations used in the development of firearms.

A source model for the proposed software system approach to the parallelization of numerical computations became the modern way of data processing concerning the so-called cloud computing, i.e. it is related to the so-called big data. The main problem to deal with relates to the distribution of a numerical computations within a computer cluster composed primarily of computer hardware commonly available on the market. It, of course, often corresponds to the state of computer equipment within a number of engineering design studios.

Since the recent attention for the big data processing in the field of computational mechanics has been pursued to the network and local computer machine based solutions related primarily to the implicit based numerical methods, e.g. in conjunction with the FETI family methods, thus the main objective of interest here has become explicit numerical methods applied in the field of massive paralell computations. Current efforts to paralelize explicit numerical methods consider local type of parallel processing on appropriate workstations requiring the presence of expensive computer hardware. The approach proposed in the thesis is able to not only exploit the possibilities of expensive high-performance hardware, but it is also able to distribute the big data of complex numerical models to the scope of a computer network which further accelerates given numerical computations.

In the developed code there was no need to use third-party libraries to create

an entire designed parallel model. It concerns to both the commercial and also to freely available software libraries. However, the use of those libraries is also possible. Since the most recent revision of C++ programming language now allows a number of required activities regarded to parallel computing, also with respect to the considerable speed of the resulting compiled code and of course with respect to the generality of such a programming language even for other purposes of software development, respectively, the mentioned programming languge was chosen as the main abstract instrument for the implementation of the respective algorithms.

During the course of the code development, various types of problems arose. The first problem which arose was the need to effectively solve the problem of general contact in an algorithmic way other than than with a native algorithm which exhibits the quadratic complexity of the algorithm. For the numerical solution of contact of bodies, the data structure called kd-tree is used. It is based on a range-searching query processed in the mentioned data structure to obtain the nodes of all the FEs that get too close to the investigated FE. Whether a given set of FE nodes gets too close to the examined FE indicates the volume of the block shape surrounding the given FE. The surrounding block shape is called a bounding box and the bounding box then indicates the range search query. One searching task is of logarithmic complexity as it is common with binary tree searching algorithms. The range-searching process is then further parallelized in terms of the set of FEs.

The second problem surfaced at a later stage of the code development. The problem relates to the redistribution of the numerical data depending on the current state of the numerical simulation over the network. In the context of the second problem, a new term Macro Entity Interaction Multigraph (MEIM) was invented. It relates to the previously dealt with problem of macro body multi-contact representing closed units of the FE meshes. The non-oriented multigraph structure is further simplified and analyzed to find the connected subgraphs representing the bulk of the data units intended to move together within the range of the considered computer network. An effort is made to provide the best balanced workload for individual workstations. The standard DFS algorithm is applied to find connected subgraphs. Multigraph assembly is based on the same principle as a detailed contact search within one FE, with the difference where the respective bounding box represents each contained body (macro entity) in the entire model.

For the numerical tests, T. Belytschko's C^0 triangular shell FE containing one integration node only, is applied. It was preferred for the low computa-

tional demands for its numerical integration, and also for the amount of data required for its definition. As an another type of nonlinear behavior of structures, including the already mentioned general contact problem, the corotational formulation for the respective triangular shell FE is considered as a source of the geometrically nonlinear behavior of structures in the range of small strains but large rotations. Consideration of geomerically non-linear behavior for such an FE is so much easier due to its geometrical definition using the first-order polynomial. The simulation results can then be further validated using the commercial software analytical tool Ansys LS-Dyna, where the respective FE is also implemented.

For the purpose of validating and also the purpose of results presentation, the Paraview software tool is used as it is a popular instrument widespread among the scientific community but also in an engineering practice. The specific result data export is executed into the files whose format is defined within the VTK.

The FEXP solver designed to test the functionality mentioned above is divided into two main parts. The first part concerns computations possible to run on a local workstation only. It uses the cores of the currently installed processor for its configurable parallel computation. The computation can also be performed in the sequential manner of an instruction execution primarily for the debug purpose of numerical computations and for performance comparison tests. It is represented by one executable only. The second part concerns the possibility of solving an appropriate numerical model in the scope of the computer network with local type of parallelization in each of the connected workstations. Such a type of solver is referred to as a hybrid-parallel FEXP solver in the text of thesis. The standard client-server software architecture is applied in the hybrid-parallel FEXP solver. The TCP/IP protocol of the transport/network layer is chosen with conjunction of Berkeley like network socket type in the Windows OS environment.

Many new features of C++ programming language ver. 14 are considered throughout the entire FEXP solver software solution. The most important improvements in the programming language are the possibility of using portable platform-independent native threads for parallel computations and the specific way of handling dynamically allocated memory through the so-called smart pointers similarly as in open source high performance 3D graphics toolkit OpenSceneGraph (OSG), e.g. it is used for the topology mapping of a civil and machinery structures in the already mentioned software tools RFEM and RSTAB developed by Dlubal Software company. A new way of dealing with the dynamically allocated memory then increases the program safety and eliminates

the problem of memory leaks. Throughout the entire source code, an emphasis is placed on the general design of the relevant functionality using templates, primarily due to later scalability for further functionality and program maintenance.

Finally, the FEXP Solver Manager was designed to simplify manipulation with the FEXP solver. It gives the ability to set up the solver, edit the solver's input data, run the solver and it also enables the monitoring of computation and the controlling of its behaviour through a graphical user interface (graphical UI or GUI). The functionality of the FEXP Solver Manager is programmed in the C# programming language using the Microsoft .NET Framework 4.6. An emphasis is placed on a modern object oriented and functional type of programming and on the asynchronous execution of a number of complex processes. It is primarily for the control of ongoing processes and for securing the responsivity of the graphical UI. The FEXP Solver Manager is ready to manage the work of both the FEXP solver focused on a single workstation as well as a hybrid-parallel type of the FEXP solver.

Given the long lasting and challenging development of such a distributed software application like the FEXP solver, only a few simpler numerical testing models have been considered. The reasons are the specifics of a multi-process execution control where completely different types of problems occur often with a random character that is natural for the such type of applications compared to common sequential running applications. The complicated random nature of program errors and the generally time-consuming nature of numerical simulations involved cause considerable complications in the time consumption for the development of the respective application.

5.2 Contributions of the Thesis

First and foremost the scientific contribution is realted to a somewhat different view of the distribution of numerical computations in nonlinear dynamics of structures provided by the explicit integration of equations of motion compared to those commonly applied approaches in the field of the parallelization of explicit numerical computations primarily focused on powerful single workstations. This refers to the so-called "big data" in terms of cloud computing, where ordinary workstations connected to a computer network are used for the application of parallel algorithms.

With respect to the effectiveness of such a solution, it was necessary to come up with a parallel algorithm competitive with those normally applied on single

powerful workstations. Therefore, a specific type of domain decomposition of the FE mesh was applied. It considers those parts of the FE mesh representing the entire integral parts of the numerical model meaning single bodies (connected subgraphs of the entire FE mesh). Those parts of the FE mesh can interact with the surrounding structures by means of the contact forces only. Direct contact interactions of the individual bodies must be solved together in one of the workstations within the considered computer network. The disadvantage of such an approach is the state, where all the bodies are in such interaction that it is no longer possible to use the given domain decomposition method and it is necessary to move all the data from the workstations within the computer network to a single one.

For the distribution of the respective model data on the network, a specific algorithm has been proposed. However, it does not consider further metadata relating to the performance of individual workstations of the heterogeneous computer cluster as well as the speed of individual connections within the computer network. It is just one of the parts providing a greater scope for further scientific research. It mainly applies to models containing a large number of interacting bodies numerically dealt with within large computer networks.

Another benefit is the parallel solution of a contact problem. Here it relates to the application of the kd-tree data structure used for such purposes. The contact search itself can then be further parallelized. It then also allows further scientific research. It also applies to the implementation of other types of searching algorithms and their subsequent performance analysis, performance tunning and improvements.

The designed architecture of the FEXP solver simply allows the implementation of new algorithms which are not only related to the themes mentioned above. Thanks to the clearly defined interface within the FEXP solver, it is possible to change the content of the individual parts under consideration. It allows the focus to purely be on the details of the specific problem solution without the need to re-develop the entire software solution. Compared to other similar software systems, the FEXP solver has been primarily designed from the outset for the purpose of implementing parallel algorithms. Therfore, it should be able to flexibly respond to the new approaches in the field of massive parallel computing.

5.3 Future Work

Referring to all of the previously mentioned topics, further development of the FEXP solution can be expected. It concerns both the implementation of new algorithms and also other hardware and software technologies. In the not too distant future, it can be assumed that the development of the FEXP solver will most likely get the capability to numerically solve even other physical phenomenas rather than just the problems related to the mechanics of solids.

However, with a view to the near future, it is necessary to continue with the implementation of the above-mentioned topics. Those have been either only partially solved or they have not already been implemented in the FEXP solver yet. Thus, the author of the dissertation thesis suggests to explore the following:

- Apply all the data structures capable of handling spatial data related to the general contact handling. This concerns both the *octree* data structure and the data structures primarily geared to spatial databases. These are concerned mainly with the various mutations of the data structure of the so-called *R-tree*, which are still the focus of current scientific research.

- Apply the NVIDIA CUDA technology in terms of the GPGPU to some parts of the computations. The first algorithms for the parallel implementation on GPUs should be algorithms related to the detection of body contacts in a complex structural environment. It applies to the algorithms mentioned in the previous item. The framework for connecting the CUDA technology to the FEXP solver is now in the final stage of completion.

- Apply some of the advanced heuristic algorithms to the MEIM analysis. Here, as a suitable candidate, a group of so-a called meta-heuristic algorithms appears. For that purpose, it may be necessary to consider additional information related to both the communication speed within the computer network and the hardware type of the respective workstations of which they are made up.

- Apply into the FEXP solver the Boost.Asio library for the network socket communication as a portable platform-independent technology. Preparations have already been made in this context.

- Apply into the FEXP solver some type of service oriented technology for netwoirk communication. As a first suitable candidate, the Microsoft

.NET/WCF technology appears. The given SOA technologies should simplify and also make more robust and secure an inter-process communication within a computer network. Such an SOA solution is then probably the most suitable candidate for the possible further commercial implementation of the mentioned distributed numerical computations.

- Apply into the FEXP solver the further type of finite elements. In particular, the spatial types of FEs that are computationally demanding in terms of the amount of the required computational operations. Thus, the spatial types of the FEs represent a suitable environment requiring the application of parallel algorithms for performance intensification. In such context, the theoretical preparation for the application of the theory of large deformations has already been carried out as a further source of nonlinear behavior of structures including nonlinear contact.

- It would be interesting to apply further constitutive material laws. Plasticity in regime of small and large strains in terms of the spatial FE seems to be very interesting in such a context.

The above listing of the goals to scientific reasearch and their implementation within the scope of the FEXP solver contains a number of an advanced and often unexplored topics. The mentioned themes are seen as those at least partially prepared for realization in the foreseeable future.

Bibliography

[1] Alamatian J.: *A new formulation for fictitious mass of the Dynamic Relaxation method with kinetic damping*, Computers & Structures, Vol. 90-91, pp. 42–54, Elsevier, 2012.

[2] Aliaga J.I., Pérez J., Quintana-Ortí E.S.: *Systematic Fusion of CUDA Kernels for Iterative Sparse Linear System Solvers*, Euro-Par 2015: Parallel Processing, 21st International Conference on Parallel and Distributed Computing, Vienna, Austria, pp. 675-686, Springer-Verlag Berlin Heidelberg, 2015.

[3] Aristotle: *Physics*, Kessinger Publishing, LLC, 2004.

[4] Bedford A.: *Hamiltons Principle in Continuum Mechanics*, John Wiley & Sons, 1986.

[5] Belytschko T., Hsieh B.J.: *Nonlinear Transient Finite Element Analysis with Convected Coordinates*, International Journal for Numerical Methods in Engineering, Vol. 7(3), pp. 255-271, John Wiley & Sons, 1973.

[6] Belytschko T., Lin J.I., Chen-Shyh Tsay: *Explicit algorithms for the nonlinear dynamics of shells*, Computer Methods in Applied Mechanics and Engineering, Vol. 42(2), pp. 225-251, Elsevier, 1984.

[7] Belytschko T., Liu W.K., Moran B., Elkhodary K.: *Nonlinear Finite Elements for Continua and Structures*, John Wiley & Sons, Second Edition, 2014.

[8] Belytschko T., Stolarski H., Carpenter N.: *A C^0 triangular plate element with one-point quadrature*, International Journal for Numerical Methods in Engineering, Vol. 20(5), pp. 787–802, John Wiley & Sons, 1984.

[9] Bentley J.L.: *Multidimensional binary search trees used for associative searching*, Communications of the ACM, Vol. 18(9), pp. 509–517, 1975.

[10] Berg M. de, Cheong O., Kreveld M. van, Overmars M.: *Computational Geometry: Algorithms and Applications*, Springer, Third Edition, 2008.

[11] Berger M.J., Bokhari S.H.: *A partitioning strategy for nonuniform problems on multiprocessors*, IEEE Transactions on Computers, Vol. C-36, No. 5, pp. 570–580, 1987.

[12] Blewett R., Clymer A.: *Pro Asynchronous Programming with .NET*, Apress, 2015.

[13] Bondy J.A., Murty U.S.R.: *Graph Theory With Applications*, Elsevier Science Ltd/North-Holland, 1976.

[14] Borst R.D., Crisfield M.A., Remmers J.J.C, Verhoosel C.V.: *Nonlinear Finite Element Analysis of Solids and Structures*, John Wiley & Sons, Second Edition, 2012.

[15] Borwein J., Zhu Q.: *Techniques of Variational Analysis (CMS Books in Mathematics)*, Springer, 2005.

[16] Bouknight W.J., Denenberg S.A., McIntyre D.E., Randall J.M, Sameh A.H., Slotnick D.L.: *The Illiac IV System*, Proceedings of the IEEE, Vol. 60, No. 4, 1972.

[17] Brenner S., Scott R.: *The Mathematical Theory of Finite Element Methods (Texts in Applied Mathematics)*, Springer, Third Edition, 2007.

[18] Brio M.: *Numerical Time-Dependent Partial Differential Equations for Scientists and Engineers*, Academic Press, 2010.

[19] Buluç A., Gilbert J.R., Budak C.: *Solving path problems on the GPU*, Parallel Computing, Elsevier, 2010.

[20] Courant R., Fredrichs K.O., Lewy H.: *On the partial difference equations of mathematical physics*, IBM Journal of Research and Development, 11 (2): 215–234, 1967.

[21] Day A.S.: *An introduction to dynamic relaxation*, The Engineers, Vol. 219, pp. 218-221, 1965.

[22] Divecchio M.C.: *The Design of the Parallel Arithmetic Unit in PEPE*, University of Pennsylvania Department of Computer and Information Science, Technical Report No. MSCIS-78-22, 1978.

[23] Dostál Z., Horák D., Kučera R.: *Total FETI - an easier implementable variant of the FETI method for numerical solution of elliptic PDE*, Communications in Numerical Methods in Engineering, Vol. 22, No. 6, pp. 1155-1162, 2006.

[24] Du P., Weber R., Luszczek P., Tomov S., Peterson G., Dongarra J.: *From CUDA to OpenCL: Towards a performance-portable solution for multi-platform GPU programming*, Parallel Computing, Vol. 38(8), pp. 391–407, Elsevier, 2012.

[25] Dunigan T.H.: *Performance of the Intel iPSC/860 and Ncube 6400 hypercubes*, Parallel Computing, Elsevier, Volume 17, Pages 1285-1302, 1991.

[26] Dziubak T., Matulewski J.: *An object-oriented implementation of a solver of the time-dependent Schrödinger equation using the CUDA technology*, Computer Physics Communications, Vol. 183(3), pp. 800-812, Elsevier, 2012.

[27] Even S.: *Graph Algorithms*, Cambridge University Press, Second Edition, 2011.

[28] Eyerman S., Eeckhout L.: *Modeling critical sections in Amdahl's law and its implications for multicore design*, ISCA '10 Proceedings of the 37th annual international symposium on Computer architecture, pp. 362-370, 2010.

[29] Farhat C., Roux F.X.: *A method of finite element tearing and interconnecting and its parallel solution algorithm*, International Journal for Numerical Methods in Engineering, Vol. 32, No. 6, pp. 1205-1227, John Wiley & Sons, 1991.

[30] Ferland K.: *Discrete Mathematics An Introduction to Proofs and Combinatorics*, Houghton Mifflin Company, 2009.

[31] Ferrando N., Gosálvez M.A., Cerdá J., Gadea R., Sato K.: *Octree-based, GPU implementation of a continuous cellular automaton for the simulation of complex, evolving surfaces*, Computer Physics Communications, Vol. 182(3), pp.628-640, Elsevier, 2011.

[32] Gamow G.: *Thirty Years that Shook Physics: The Story of Quantum Theory*, Dover Publications, Revised ed. edition, 1985.

[33] Gaster B., Howes L., Kaeli D.R., Mistry P., Schaa D.: *Heterogeneous Computing with OpenCL*, Morgan Kaufmann, 2011.

[34] Goldstein H., Poole Ch., Safko J.: *Classical Mechanics: Pearson New International Edition*, Pearson Education, Third Edition, 2001.

[35] Gorobets A.V., Trias F.X., Oliva A.: *A parallel MPI + OpenMP + OpenCL algorithm for hybrid supercomputations of incompressible flows*, Computers & Fluids, Vol. 88, pp. 764–772, Elsevier, 2013.

[36] Gropp W., Lusk E., Skjellum A.: *Using MPI - 2nd Edition: Portable Parallel Programming with the Message Passing Interface (Scientific and Engineering Computation)*, The MIT Press, Second edition, 1999.

[37] Gruber P., Zeman J., Kruis J., Šejnoha M.: *Homogenization of composites with interfacial debonding using duality-based solver and micromechanics*,

Computer Assisted Mechanics and Engineering Sciences, 16(1): 59-76, 2009.

[38] Guan Q.: *Parallel Algorithms for Geographic Processing*, Dissertation, University of California, Santa Barbara, 2008.

[39] Hallquist J.O.: *LS-DYNA Theoretical Manual*, Livermore: Livermore Software Technology Corporation, 1998.

[40] Halphen B., Nguyen Q.S: *Sur les matériaux standards généralisés*, Journal de Mécanique, Vol. 14, pp. 39–63, 1975.

[41] Har J., Tamma K.: *Advances in Computational Dynamics of Particles, Materials and Structures*, John Wiley & Sons, 2012.

[42] Hart J. M.: *Windows System Programming (Addison-Wesley Microsoft Technology Series)*, Addison-Wesley Professional, 2015.

[43] Hertz H.: *Über die Berührung fester elastischer Körper*, Journal für die Reine und Angewandte Mathematik, Vol. 29, pp. 156–171, 1882.

[44] Hillis W.D.: *The Connection Machine (Mit Press Series in Artificial Intelligence)*, The MIT Press, 1986.

[45] Hlaváček I., Haslinger J., Nečas J., Lovíšek J.: *Solution of Variational Inequalities in Mechanics (Applied Mathematical Sciences)*, Springer, 1988.

[46] Holuša L., Kratochvíl J., Křížek M., Marek I. Ženíšek A.: *Miloš Zlámal, zakladatel matematické teorie metody konečných prvků*, VUTIUM, 2006.

[47] Hopcroft J., Tarjan R.: *Algorithm 447: Efficient algorithms for graph manipulation*, Communications of the ACM, Vol. 16(6), pp. 372-378, ACM, 1973.

[48] Hwang K., Dongarra J., Fox G.C.: *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*, Morgan Kaufmann, 2011.

[49] Chapman S.: *Fortran 95/2003 for Scientists and Engineers*, McGraw-Hill Education, Third Edition, 2007.

[50] Chaves E.W.V: *Notes on Continuum Mechanics (Lecture Notes on Numerical Methods in Engineering and Sciences)*, Springer, 2013.

[51] Chen W., Beister M., Kyriakou Y., Kachelries M.: *High performance median filtering using commodity graphics hardware*, Nuclear Science Symposium Conference Record (NSS/MIC) IEEE, pp. 4142-4147, 2009.

[52] Christensen J., Bastien Ch.: *Nonlinear Optimization of Vehicle Safety Structures: Modeling of Structures Subjected to Large Deformations*, Butterworth-Heinemann, 2015.

[53] Idelsohn S.R., Marti J., Becker P., Oñate E.: *Analysis of multifluid flows with large time steps using the particle finite element method*, International Journal for Numerical Methods in Fluids, Vol. 75, pp. 621–644, John Wiley & Sons, 2014.

[54] Irvine K.R.: *Assembly Language for Intel-Based Computers*, Prentice Hall, 2007.

[55] Jacobsen D.A., Senocak I.: *Multi-level parallelism for incompressible flow computations on GPU clusters*, Parallel Computing, Vol. 39, pp. 1-20, Elsevier, 2013.

[56] Johnsen S.F., Taylor Z.A., Clarkson M.J., et al.: *NiftySim: A GPU-based nonlinear finite element package for simulation of soft tissue biomechanics*, International Journal of Computer Assisted Radiology and Surgery, Vol. 10(7), pp. 1077–1095, Springer, 2015.

[57] Johnson M.W., Amin M.H.S., Gildert S., Lanting T., Hamze F., Dickson N., Harris R., Berkley A.J., Johansson J., Bunyk P., Chapple E.M., Enderud C., Hilton J.P., Karimi K., Ladizinsky E., Ladizinsky N., Oh T., Perminov I., Rich C., Thom M.C., Tolkacheva E., Truncik C.J.S, Uchaikin S., Wang J., Wilson B., at al: *Quantum annealing with manufactured spins*, Nature, Volume 473, pp. 194-198, 2011.

[58] Kennedy J.M., Belytschko T., Lin J.I.: *Recent developments in explicit finite element techniques and their application to reactor structures*, Nuclear Engineering and Design, Vol. 97(1), pp. 1-24, Elsevier, 1986.

[59] Kernighan B.W., Lin S.: *An efficient heuristic procedure for partitioning graphs*, The Bell System Technical Journal, Vol. 49, No. 2, pp. 291–307, 1970.

[60] Kernighan B.W., Ritchie D.M: *The C Programming Language*, Prentice-Hall, 1978.

[61] Khoei A.R.: *Extended Finite Element Method: Theory and Applications (Wiley Series in Computational Mechanics)*, John Wiley & Sons, 2015.

[62] Krause R., Rank E.: *A fast algorithm for point-location in a finite element mesh*, Computing, Vol. 57(1), pp. 49-62, Springer-Verlag, 1996.

[63] Kruis J.: *Domain Decomposition Methods for Distributed Computing (Saxe-Coburg Publications on Computational Engineering)*, Saxe-Coburg Publications, 2007.

[64] Kruis J., Bittnar Z.: *Reinforcement-matrix interaction modeled by FETI method*, Domain Decomposition Methods in Science and Engineering XVII, Lecture Notes in Computational Science and Engineering, Vol. 60, pp. 567-573, Springer Berlin Heidelberg, 2008.

[65] Kruis J., Zeman J., Gruber P.: *Model of Imperfect Interfaces in Composite Materials and Its Numerical Solution by FETI Method*, Lecture Notes in Computational Science and Engineering, Vol. 91, pp. 337-344, 2013.

[66] Kurose J.F., Ross K.W.: *Computer Networking: A Top-Down Approach*, Pearson, Sixth Edition, 2012.

[67] Ladd T.D., Jelezko F., Laflamme R., Nakamura Y., Monroe Ch., O'Brien J.L: *Quantum Computing*, Nature, Volume 464, pp. 45-53, 2010.

[68] Lafontaine N.M., Rossi R., Cervera M. et al.: *Explicit mixed strain-displacement finite element for dynamic geometrically non-linear solid mechanics*, Computational Mechanics, Vol. 55, pp. 543–559, Springer Berlin Heidelberg, 2015.

[69] Lamport L.: *Computer: A History of the Information Machine (The Sloan Technology Series)*, Westview Press, Third Edition, 2013.

[70] Langr D., Tvrdík P., Dytrych T., Draayer J.P.: *Algorithm 947: Paraperm - Parallel Generation of Random Permutations with MPI*, ACM Trans. Math. Softw., Vol. 41(1), pp. 5:1-5:26, 2014.

[71] Langr D., Tvrdík P., Šimeček I., Dytrych T.: *Downsampling Algorithms for Large Sparse Matrices*, International Journal of Parallel Programming, Vol. 43(5), pp. 679–702, Springer, 2015.

[72] Laursen T.A.: *Computational Contact and Impact Mechanics: Fundamentals of Modeling Interfacial Phenomena in Nonlinear Finite Element Analysis*, Springer, 2002.

[73] Lea H.P., Cambierb J.-L., Cole L.K.: *GPU-based flow simulation with detailed chemical kinetics*, Computer Physics Communications, Vol. 184(3), pp. 596–606, Elsevier, 2013.

[74] LeBlanc T.J., Scott M.L., Brown C.M.: *Large-Scale Parallel Programming: Experience with the BBN Butterfly Parallel Processor*, Butterfly

Project Report, Computer Science Department, University of Rochester, 1988.

[75] Lefebvre M., Guillen P., Le Gouez J.-M., Basdevant C.: *Optimizing 2D and 3D structured Euler CFD solvers on Graphical Processing Units*, Computers & Fluids, Vol. 70, pp. 136-147, Elsevier, 2012.

[76] Li R., Saad Y.: *GPU-accelerated preconditioned iterative linear solvers*, The Journal of Supercomputing, Vol. 63, pp. 443-466, Springer US, 2013.

[77] Lipschutz S., Lipson M.: *Schaum's Outline of Discrete Mathematics*, McGraw-Hill, Third Edition, 2007.

[78] Liu W., Schmidt B., Voss G., Müller-Wittig W.: *Accelerating molecular dynamics simulations using Graphics Processing Units with CUDA*, Computer Physics Communications, Vol. 179(9), pp. 634–641, Elsevier, 2008.

[79] Lucas É.: *Récreations Mathématiques (Vol. I)*, Paris:Gauthier-Villars, 1882.

[80] Manolopoulos Y., Nanopoulos A., Papadopoulos A.N., Theodoridis Y.: *R-Trees: Theory and Applications*, Advanced Information and Knowledge Processing, Springer, 2006.

[81] Matoušek J., Nešetřil J.: *Kapitoly z diskrétní matematiky*, Karolinum, 2010.

[82] Maugin G.A.: *Continuum Mechanics Through the Twentieth Century: A Concise Historical Perspective (Solid Mechanics and Its Applications)*, Springer, 2013.

[83] Maurer D.: *Energy Expressions in Explicit Finite-Element Methods*, 2003.

[84] Mehlhorn K.: *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry*, Monographs in Theoretical Computer Science. An EATCS Series, Springer, 1990.

[85] Melenka J.M., Babuška I.: *The partition of unity finite element method: Basic theory and applications*, Computer Methods in Applied Mechanics and Engineering, Vol. 139(1-4), pp. 289-314, Elsevier, 1996.

[86] Mielke A.: *A Mathematical Framework for Generalized Standard Materials in the Rate-Independent Case*, Lecture Notes in Applied and Computational Mechanics, Vol. 28, pp. 399-428, 2006.

[87] Nagel Ch., Glynn J.: *Professional C# 5.0 and .NET 4.5.1*, Wrox, 2014.

[88] Naumov M.: *Parallel Solution of Sparse Triangular Linear Systems in the Preconditioned Iterative Methods on the GPU*, NVIDIA Technical Report NVR-2011-001, NVIDIA, 2011.

[89] Neal M.O., Belytschko T.: *Explicit-explicit subcycling with non-integer time step ratios for structural dynamic systems*, Computers & Structures, Vol. 31(6), pp. 871-880, Elsevier, 1989.

[90] Nečas J.: *Les methodes directes en theorie des equations elliptiques*, Academia, Praha, and Masson et Cie, Editeurs, Paris, 1967.

[91] Němec I., Kolář V., Ševčík I., Vlk Z., Blaauwendraat J., Buček J., Teplý B., Novák D., Štembera V.: *Finite Element Analysis of Structures, Principles and Praxis*, Shaker Verlag, 2010.

[92] Ng S., Leung K.: *Induction of Quadratic Decision Trees using Genetic Algorithms and k-D Trees*, 2003.

[93] Oller S.: *Nonlinear Dynamics of Structures (Lecture Notes on Numerical Methods in Engineering and Sciences)*, Springer, 2015.

[94] Oñate E.: *Structural Analysis with the Finite Element Method. Linear Statics: Volume 1: Basis and Solids (Lecture Notes on Numerical Methods in Engineering and Sciences)*, Springer, 2009.

[95] Oñate E.: *Structural Analysis with the Finite Element Method. Linear Statics: Volume 2: Beams, Plates and Shells (Lecture Notes on Numerical Methods in Engineering and Sciences)*, Springer, 2013.

[96] Oyarzun G., Borrell R., Gorobets A., Oliva A.: *MPI-CUDA sparse matrix–vector multiplication for the conjugate gradient method with an approximate inverse preconditioner*, Computers & Fluids, Vol. 90, pp. 244-252, Elsevier, 2013.

[97] Parsons B., Wilson E.A.: *A Method for Determining the Surface Contact Stresses Resulting From Interference Fits*, Journal of Engineering for Industry, Vol. 92(1), pp. 208-218, 1970.

[98] Patzák B.: *OOFEM - an object-oriented simulation tool for advanced modeling of materials and structures*, Acta Polytechnica, 52(6):59–66, 2012.

[99] Patzák B., Rypl D., Kruis J.: *MuPIF – A distributed multi-physics integration tool*, Advances in Engineering Software, Vol. 60-61, pp. 89-97, Elsevier, 2012.

[100] Perrot G., Domas S., Couturier S.: *Fine-tuned high speed implementation of a gpu-based median filter*, Journal of Signal Processing Systems, pp. 1-6, 2011.

[101] Pickeringa B.P., Jacksona Ch.W., Scoglandb T.R.W., Wu-Chun Fengb, Roya Ch.J.: *Directive-based GPU programming for computational fluid dynamics*, Computers & Fluids, Vol. 114, pp. 242–253, Elsevier, 2015.

[102] Reddy J.N.: *An Introduction to Nonlinear Finite Element Analysis: with applications to heat transfer, fluid mechanics, and solid mechanics*, Oxford University Press, Second Edition, 2015.

[103] Reissner E.: *On transverse bending of plates, including the effect of transverse shear deformation*, International Journal of Solids and Structures, Vol. 11(5), pp. 569-573, Elsevier, 1975.

[104] Rek V., Němec I.: *Parallel Computing Procedure for Dynamic Relaxation Method on GPU Using NVIDIA's CUDA*, Applied Mechanics and Materials, Vol. 821, pp. 331-337, Trans Tech Publications, 2016.

[105] Rek V., Němec I.: *Parallel Computation on Multicore Processors Using Explicit Form of the Finite Element Method and C++ Standard Libraries*, Strojnícky casopis – Journal of Mechanical Engineering, Vol. 66(2), pp. 67-78, De Gruyter, 2016.

[106] Rodriguez J., Rio R., Cadou J.M., Troufflard J.: *Numerical study of dynamic relaxation with kinetic damping applied to inflatable fabric structures with extensions for 3D solid element and non-linear behavior*, Thin-Walled Structures, Vol. 49(11), pp. 1468-1474, Elsevier, 2011.

[107] Rodrígueza M., Blesab F., Barrio R.: *OpenCL parallel integration of ordinary differential equations: Applications in computational dynamics*, Computer Physics Communications, Vol. 192, pp. 228–236, Elsevier, 2015.

[108] Rosen K.H.: *Handbook of Discrete and Combinatorial Mathematics, Second Edition (Discrete Mathematics and Its Applications)*, CRC Press, 2000.

[109] Ross A.: *A Rudimentary History of Dynamics*, Modeling, Identification and Control, Vol. 30(4), pp. 223–235, Norwegian Society of Automatic Control, 2009.

[110] Říha L., Brzobohatý T., Markopoulos A., Jarošová M., Kozubek T.: *Implementation of hybrid total FETI (HTFETI) solver for multi-core archi-*

tectures, AIP Conference Proceedings, Vol. 1648, American Institute of Physics, 2015.

[111] Salvadorea F., Bernardinib M., Botti M.: *GPU accelerated flow solver for direct numerical simulation of turbulent flows*, Journal of Computational Physics, Vol. 235, pp. 129–142, Elsevier, 2013.

[112] Samet H.: *Foundations of Multidimensional and Metric Data Structures*, The Morgan Kaufmann Series in Computer Graphics, Morgan Kaufmann, 2006.

[113] Santos A., Teixeira J.M., Farias T., Teichrieb V., Kelner J.: *Understanding the Efficiency of kD-tree Ray-Traversal Techniques over a GPGPU Architecture*, International Journal of Parallel Programming, Vol. 40(3), pp. 331–352, Springer, 2011.

[114] Sedgewick R., Flajolet P.: *An Introduction to the Analysis of Algorithms*, Addison-Wesley Professional, 2013.

[115] Shreiner D., Sellers G., Kessenich J.M., Licea-Kane B.: *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3*, Addison-Wesley Professional, Eighth Edition, 2013.

[116] Signorini A.: *Sopra alcune questioni di elastostatica*, Atti della Societa Italiana per il Progresso delle Scienze, 1933.

[117] Simon H.D.: *Parallel computational fluid dynamics: implementations and results*, Scientific and engineering computation, The MIT Press, Cambridge, MA, 1992.

[118] Skiena S.S.: *The Algorithm Design Manual*, Springer, 2008.

[119] Sosutha S., Mohana D.: *Heterogeneous Parallel Computing Using Cuda for Chemical Process*, Procedia Computer Science, Vol. 47, pp. 237-246, Elsevier, 2015.

[120] Souza Neto E.A. de, Peric D., Owen D.R.J.: *Computational Methods for Plasticity: Theory and Applications*, John Wiley & Sons, 2008.

[121] Stallings W.: *Operating Systems: Internals and Design Principles*, Prentice Hall, Sixth Edition, 2008.

[122] Storaasli O.O., Peebles S.W., Crockett T.W., Knott J.D., Adams L.: *The finite element machine: An experiment in parallel processing*, Technical Report, NASA Langley Research Center; Hampton, VA, United States, 1982.

[123] Strömberg N., Johansson L., Klarbring A.: *Derivation and analysis of a generalized standard model for contact, friction and wear*, International Journal of Solids and Structures, Vol. 33, pp. 1817-1836, 1996.

[124] Tabarrok C., Rimrott F.P.: *Variational Methods and Complementary Formulations in Dynamics (Solid Mechanics and Its Applications)*, Springer, 1994.

[125] Vallée C., Lerintiu C., Fortuné D., Atchonouglo K, Ban M.: *Representing a non-associated constitutive law by a bipotential issued from a Fitzpatrick sequence*, Archives of Mechanics, Vol. 61, pp. 325-340, 2009.

[126] Walshaw C., Cross M.: *JOSTLE: parallel multilevel graph-partitioning software—an overview*, Mesh Partitioning Techniques and Domain Decomposition Techniques, pp. 27–58, Civil-Comp Press, 2007.

[127] Wang Y.-X, Zhang L.-L., Liu W., Che Y.-G., Xu C.-F., Wang Z.-H., Zhuang Y.: *Efficient parallel implementation of large scale 3D structured grid CFD applications on the Tianhe-1A supercomputer*, Computers & Fluids, Vol. 80, pp. 244-250, Elsevier, 2013.

[128] Wei Z., Jang B., Jia Y: *A fast and interactive heat conduction simulator on GPUs*, Procedia Computer Science, Vol. 270, pp. 496-505, Elsevier, 2014.

[129] Weller R.: *New Geometric Data Structures for Collision Detection and Haptics*, Springer Series on Touch and Haptic Systems, Springer, 2013.

[130] Wiener N.: *Cybernetics: or control and communication in the animal and the machine*, The MIT Press, Cambridge, MA, 1948.

[131] Wilson E.A., Parsons B.: *Finite element analysis of elastic contact problems using differential displacement*, International Journal for Numerical Methods in Engineering, Vol. 2(3), pp. 387-395, John Wiley & Sons, 1970.

[132] Wriggers P.: *Computational Contact Mechanics*, Springer, Second Edition, 2006.

[133] Wriggers P.: *Nonlinear Finite Element Methods*, Springer, 2008.

[134] Wriggers P., Zavarise G.: *A formulation for frictionless contact problems using a weak form introduced by Nitsche*, Computational Mechanics, Vol. 41(3), pp. 407–420, Springer, 2008.

[135] Wright R.S., Haemel N., Sellers G., Lipchak B.: *OpenGL SuperBible: Comprehensive Tutorial and Reference*, Addison-Wesley Professional, Fifth Edition, 2010.

[136] Wu S.R., Gu L.: *Introduction to the Explicit Finite Element Method for Nonlinear Transient Dynamics*, John Wiley & Sons, 2012.

[137] Zeman J., Gruber P.: *Numerical approach to a rate-independent model of decohesion in laminated composites*, Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar, pp. 239-250, Institute of Mathematics AS CR, 2010.

[138] Zhou K., Hou R.Q., Wang B., Guo B.: *Real-Time KD-Tree Construction on Graphics Hardware*, ACM Transactions on Graphics (TOG), Vol. 27, pp. 1-11, 2008.

[139] Zlámal M.: *On the finite element method*, Numerische Mathematik, Vol. 12(5), pp. 394-409, Springer, 1968.

Publications of the Author

Relevant Publications of the Author

[A.1] Rek V., Němec I. *Parallel Computing Procedure for Dynamic Relaxation Method on GPU Using NVIDIA's CUDA*. Engineering Mechanics 2015 - 21st International Conference: Extended Abstracts, May 11 - 14, 2015, Svratka, Czech Republic, Institute of Theoretical Applied Mechanics, Academy of Science of the Czech Republic, v.v.i., Prague, 2015, ISBN: 978-80-86246-42-0, ISSN: 1805-8248.

[A.2] Rek V., Němec I. *Parallel Computing Procedure for Dynamic Relaxation Method on GPU Using NVIDIA's CUDA*. Applied Mechanics and Materials, Vol. 821, pp. 331-337, Trans Tech Publications, Switzerland, 2016, https://doi.org/10.4028/www.scientific.net/AMM.821.331.

The paper has been cited in:

- Guo S., Cai X., Gao B., Yuhua J. *An improved VR training system for vascular interventional surgery*, 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1667, 2016.

[A.3] Rek V., Němec I. *Parallel Computation on Multicore Processors Using Explicit Form of the Finite Element Method and C++ Standard Libraries*. Applied Mechanics 2016 - 18th International Scientific Conference, Extended Abstracts, April 11 - 13, 2016, Banská Štiavnica, Slovak Republic, Vydavateľstvo STU, Bratislava, 2016, ISBN: 978-80-227-4547-5.

[A.4] Rek V., Němec I. *Parallel Computations and C++ Standard Libraries*. Engineering Mechanics 2016 - 22st International Conference: Conference Proceedings, May 9 - 12, 2016, Svratka, Czech Republic, Institute of Theoretical Applied Mechanics, Academy of Science of the Czech Republic, v.v.i., Prague, 2016, ISBN: 978-80-87012-59-8, ISSN: 1805-8248.

[A.5] Rek V., Němec I. *Parallel Computation on Multicore Processors Using Explicit Form of the Finite Element Method and C++ Standard Libraries*. Strojnícky casopis – Journal of Mechanical Engineering, Vol. 66(2), pp. 67-68, De Gruyter Open, 2016, https://doi.org/10.1515/scjme-2016-0020.

Remaining Publications of the Author

[A.6] Němec I., Trcala M., Rek V. *Nelineární mechanika.* Vutium Publisher, Czech Republic, 2018, ISBN: 978-80-214-5519-1.

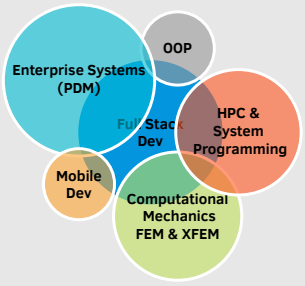
Václav Rek

ICT Developer

LinkedIn /in/hsgadgil

rekv@seznam.cz

Skills



Interests

Enterprise Systems (PDM)

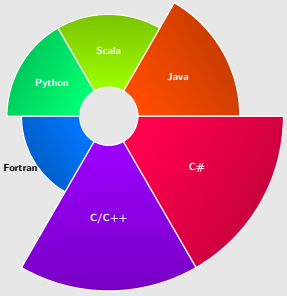
Mobile Dev

Computational Mechanics, FEM & XFEM

HPC & System Programming

OOP

Languages



Education

2012 - 2018 (Expected) **Ph.D., Civil Engineering** — Brno University of Technology
Brno, Czech Republic
Doctoral degree study program: *Structures and Traffic Constructions*
Doctoral thesis: *The Exploitation of Parallelization to Numerical Solutions Regarding Problems in Nonlinear Dynamics*

2002 - 2007 **MSc., Civil Engineering** — Brno University of Technology
Brno, Czech Republic
Master degree study program: *Water Management and Water Structures*
Master thesis: *Risk Analysis of Water Treatment Technological Processes*

Research

2012 - 2018 **Ph.D. Research Student** — Brno University of Technology
Parallelization of FEM based on the combination regarding computer network (Big Data) and local machine specific hardware resource accessibility (CPU, GPU)—High Performance Computing (HPC), explicit form of FEM, computational dynamics, fracture mechanics (XFEM), nonlinear geometrical analysis in mechanics of deformable bodies (large deformations, contact mechanics, etc.), robotics (simulation and control), theoretical computer science, etc.
Publications:

- Rek V., Němec I.: *Parallel Computing Procedure for Dynamic Relaxation Method on GPU Using NVIDIA's CUDA*, Applied Mechanics and Materials, Vol. 821, pp. 331-337, Trans Tech Publications, Switzerland, 2016, `https://doi.org/10.4028/www.scientific.net/AMM.821.331`
- Rek V., Němec I.: *Parallel Computation on Multicore Processors Using Explicit Form of the Finite Element Method and C++ Standard Libraries*, Strojnícky casopis – Journal of Mechanical Engineering, Vol. 66(2), pp. 67-68, De Gruyter Open, 2016, `https://doi.org/10.1515/scjme-2016-0020`
- Němec I., Trcala M., Rek V.: *Nelineární mechanika*, Vutium Publisher, Czech Republic, 2018, ISBN: 978-80-214-5519-1

Experience

May 2018 - Present **Scientific Software Developer** — FEM consulting
Computational mechanics, computer science, etc.

March 2015 - May 2018 **.NET/C# Enterprise Developer** — CADTeam
Development of *Product Data Management system (PDM)* with *.NET/WCF* technology for *SOA*. Concurrent asynchronous communication, Windows system programming, technical analysis and more.

Aug 2010 - March 2015 **C/C++ Developer** — Dlubal Software
General development in *C++*, optimization of legacy *Fortran* codes, *MFC* and *Qt* technology, *Python*, *Java SE*, technical analysis and more.

Dec 2007 - Aug 2010 **Software QA Tester** — Dlubal Software
Testing, composition of testing tools for automatic testing, support for customers.