Karlsruhe Institute of Technology

Proceedings of the 4th EMCL-Workshop ”Numerical Simulation, Optimization and High Performance Computing”

Vincent Heuveline

Gudrun Thäter

(Eds.)

No. 2011-12

Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)

KIT – University of the State of Baden-Wuerttemberg and
National Research Center of the Helmholtz Association

www.emcl.kit.edu

Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)
ISSN 2191–0693
No. 2011-12

Impressum

Karlsruhe Institute of Technology (KIT)
Engineering Mathematics and Computing Lab (EMCL)

Fritz-Erler-Str. 23, building 01.86
76133 Karlsruhe
Germany

KIT – University of the State of Baden Wuerttemberg and
National Laboratory of the Helmholtz Association

Proceedings of the 4th EMCL-Workshop

Numerical Simulation, Optimization and High Performance Computing

Vincent Heuveline, Gudrun Thäter
(Eds.)

Contents

Foreword

The Engineering Mathematics and Computing Lab (EMCL) was being initiated as a new institution of the Karlsruhe Institute of Technology (KIT) in July 2009. The main goal of EMCL is not only to address complex problems with high impact in the society but also to offer a platform for interdisciplinary work with academic institutions and industrial partners. With respect to the methodology, EMCL relies on a holistic approach that encompasses mathematical modeling, numerical simulation, optimization, high performance computing and cloud computing. The application topics range from energy research, meteorology and environmental research to medical engineering and biotechnology.

During the 4th-Workshop “Numerical Simulation, Optimization and High Performance Computing” (Dagstuhl, March 2011), these topics have been addressed in the context of a wide range of applications. The scientific depth of the contributions clearly show that the institution EMCL is now mature and is in a position to achieve and carry out the aspired goals. Most important however is to stress that all participants have clearly shown not only their ability to address highly complex scientific problems but also that they were able to bring their ideas into life also by means of a vital team spirit and team work. I believe that this is a key ingredient for the success of this workshop and hope that these proceedings will allow the reader to participate in the extremely exciting developments at EMCL triggered by highly innovative and motivated mathematicians.

Prof. Dr. Vincent Heuveline
Director EMCL

Efficient Data Management and Visualization for Ensemble Weather Forecast Computations

Hartwig Anzt

Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
hartwig.anzt@kit.edu

Abstract. To achieve a higher accuracy of weather predictions, an increasing number of forecasts is based on ensemble simulations. Due to the high complexity of ensemble systems and the large amount of data, specific routines which are able to pre- and post-process the simulation data in reasonable time are required. The challenges in this context include efficient data management and parallel visualization techniques.

Keywords. COSMO, Ensemble Forecast Model, Data-Management, Visualization

1 Ensemble Forecasting

Despite the fact that the underlying mathematical models as well as the numerical algorithms generating weather forecasts have improved a lot during the last decades, the simulation outcome still often differs considerably from the actual weather evolution. This has its reason not in the inexactness of the underlying models, which additionally triggers variations, but rather in the properties of the partial differential equations the model is based on, and the uncertainty of the initial data. In chaos theory, this behaviour is also known as the ”Butterfly Effect”. It generally describes the concept of sensitive dependence on initial conditions: small differences in the initial condition of a dynamical system may produce large variations in the long term behavior of the system [1].

To account for this uncertainty, stochastic or ”ensemble” forecasting is used. The basic idea is to run, in addition to the main forecast, further simulations, that are based on slightly modified data, see Fig. 1.

The described ensemble forecast can be expanded furthermore to a ”multilevel ensemble system”, by repeating the simulation not only with different initial data, but also by using different driving models, which generate the boundary conditions for the simulation. Since this leads to a high number of ensemble members in total, not only the computational cost of the simulation, but also the management and visualization of data poses a challenge.

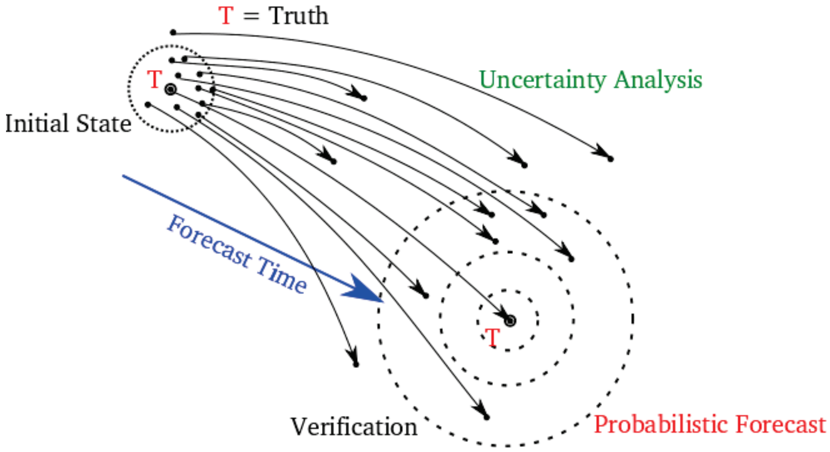


Fig. 1. An ensemble prediction system. Small Pertubations in the starting values lead to very different simulation results.

2 Data Management

Since the COSMO model [2] which is used for the multilevel forecast simulation is very memory demanding, efficient implementations require for fast memory access. Large HPC systems usually offer different storage types, with different characteristics concerning lifetime, capacity and read/write performance. Analyzing these characteristics for example for the HPC-system HC3 [3] and the connected file system LUSTRE containing the different storage areas $TMP, $HOME and $WORK (see Fig. 2 and Table 1), allows to derive rules for efficient data handling on this cluster:

- Copy data just before simulation start from data-server to file system.
- Use of $WORK instead of $HOME for faster data access.
- Real-time forward-copying of simulation data to data-server.

Property	**$TMP**	**$HOME**	**$WORK**
lifetime	batchjob	permanent	7 days
capacity	129/673/825 GB	76 TB	203 TB
backup	no	yes	no
read performance/node	70/250 MB/s	600 MB/s	1800 MB/s
write performance/node	80/390 MB/s	700 MB/s	1800 MB/s

Table 1. Characteristics of the HC3-cluster connected to LUSTRE [3].

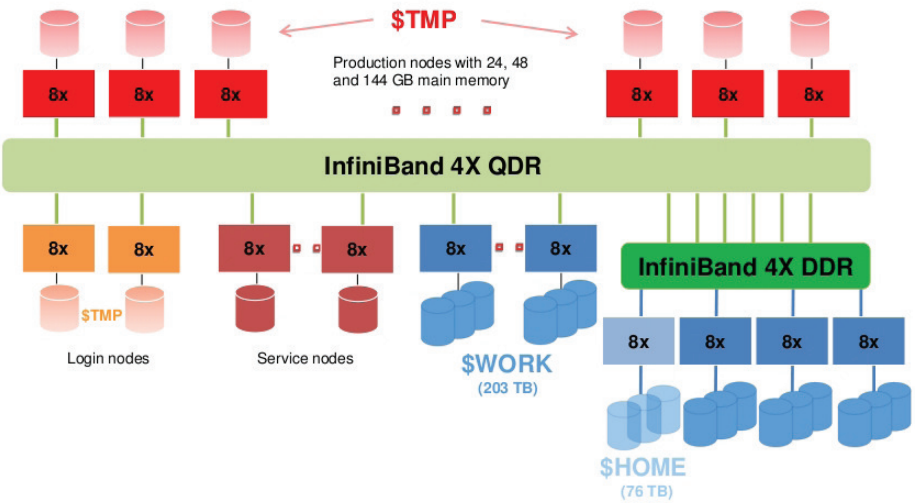


Fig. 2. Topology of HC3 and connected filesystem LUSTRE [4].

3 Visualization

To simplify the scientific analysis of the forecasts, an efficient visualization of the simulated data is essential. This includes two steps of post-processing. First, the needed variables have to be identified in the simulation data written to a binary file, and converted to a suitable format. Then, visualization tools like Paraview [5], enable the user to display the simulated data. While the efficient variable identification and data conversion is already possible by using a script written by Leonhard Scheck, scientific researcher at the *Institute for Meteorology and Climate Research, KIT*, the visualization using the Paraview interface is still slow and inefficient. Paraview scripting, a Python [6] extension able to process VTK data, offers an efficient solution for visualizing specific variables in a certain simulation state [7]. Since the simulated forecast states are written in distinct binary-files, the conversion into VTK-files and the visualization of the data using Paraview scripting can be parallelized efficiently. The generated screenshots can be merged into a forecast animation.

In the context of the multilevel-ensemble-forecast, a Python-script was developed, that efficiently merges the ensemble forecast simulation, the data handling and the visualization process into one service. The user sends the request for a forecast variable he wants to analyze to the HPC-system, which performs the simulation using the COSMO-model. The simulation data is automatically copied to the file system, and distributed to the visualization cluster. There, the demanded variables in the binary-files are identified, converted to the VTK format, and Paraview scripting is applied to visualize them. Finally, the data is sent back to the file system, merged into a forecast animation, and delivered to the user. The accelerated and simplified simulation process (see Fig. 3), allows the scientific researcher to generate more data and to expand the number of ensemble members in the multilevel ensemble system.

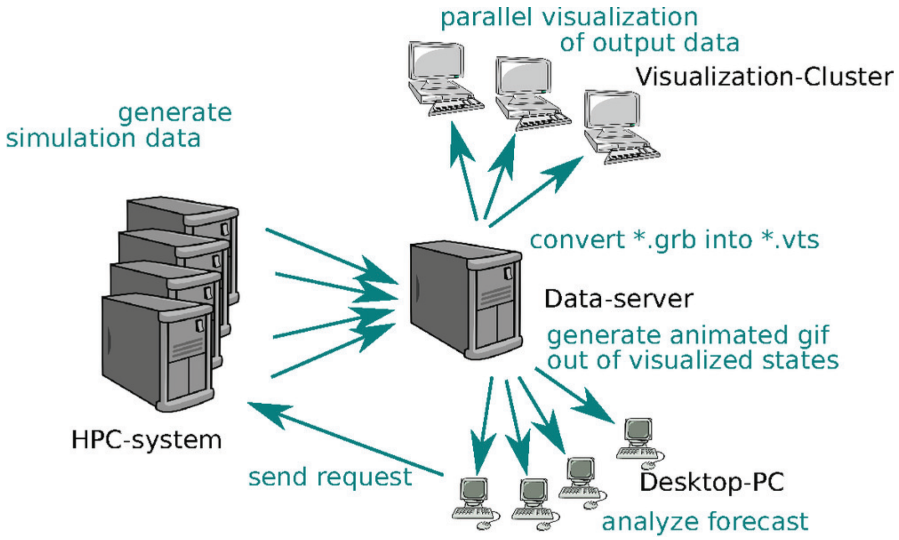


Fig. 3. Parallel data processing using a Visualization Cluster.

References

1. Lorenz, E.N.: Deterministic Nonperiodic Flow. Journal of the Atmospheric Sciences **20** (1962) 130–141
2. Schaettler, U., Doms, G., C., S.: A Description of the Nonhydrostatic Regional COSMO-Model. **4.11** (2009)
3. Webpage: (http://www.scc.kit.edu/dienste/hc3.php)
4. Wang, F., Oral, S., Shipman, G., Drokin, O., Wang, T., Huang, I.: UNDERSTANDING LUSTRE FILESYSTEM INTERNALS. National Center for Computational Sciences (2009)
5. Webpage: (http://www.paraview.org/)
6. Webpage: (http://www.python.org/)
7. Cedilnik, A., Geveci, B., Moreland, K., Ahrens, J.A., Favre, J.: Remote Large Data Visualization in the ParaView Framework. Eurographics Symposium on Parallel Graphics and Visualization (2006)

Generalized Stencil Computation on CPUs and GPUs

Werner Augustin

Shared Research Group New Frontiers in High Performance Computing Exploiting Multicore and Coprocessor Technology
Engineering Mathematics and Computing Lab (EMCL) – Karlsruhe Institute of Technology (KIT), Germany, Werner.Augustin@kit.edu

Keywords. FEM, stencil computation, CPU, GPU

1 Introduction

Discretizations of PDEs that arise in several field of natural science usually lead to large, linear systems with very sparse system matrices. This finally leads to frequent sparse matrix vector multiplications with a low computational intensity which on current hardware architectures run at fractions of the possible peak performance. Flexible use of constant coefficient stencil computations derived from finite element methods could overcome this memory bandwidth bottleneck and considerably improve overall performance.

2 Computational Intensity

Computational intensity is defined as the ratio of the number of floating operations and necessary data transfers from memory. In this brief overview article we restrict ourself to double precision floating point operations and count only the expensive transfers from and to main memory, considering cache accesses as desirable and cheap. Additionally, for the theoretical estimations we count accesses to input and output values only once, assuming some suitable tiling or domain decomposition scheme.

On the algorithmic side we have for sparse matrix vector multiplication an approx. intensity value of 1.33, independent of the original method, the matrix was gained. Multiplications using stencil on the other hand vary between approx. 23 and 51 for Laplace problems with linear or 2nd order finite element discretization up to approx. 161 for 3-dim. 2nd order FEM on elasticity problems.

Unfortunately, current hardware has a much higher floating point peak performance than memory bandwidth. This results in a necessary computational intensity of roughly 32 for Intel Nehalem to get peak performance, i.e. they run at only a fraction of it when doing a sparse matrix vector multiplication. NVIDIAs Tesla has a necessary intensity of approx. 6.5. But this is mainly due to its lower double precision performance, the current Fermi architecture also reaching the region of 30.

3 Unstructured vs. Structured Grids

Unstructured grids are traditionally used more frequently for FEM schemes. They allow a larger flexibility in the modeling of the problem domain and can be locally refined to adapt to the desired precision. From the programming point of view they allow for higher modularity since they usually they are transformed in some general sparse system matrix and left to a iterative solver that is completely agnostic of the original problem. But as mentioned above, solving these systems can only be performed at a fraction of the hardware peak performance (see [1] for multicore architectures and [2] for performance on NVIDIA GPUs).

Structured grids on the other hand are also in use for quite a long time, mostly for finite difference schemes, were stencil computations can be used easily. But of course they can also be extended for FEM. There they could considerably lower the memory bandwidth pressure and with their more regular memory access could be easier adapted to modern and future hardware. (see [3] for results on multicore architectures and [4] on GPUs).

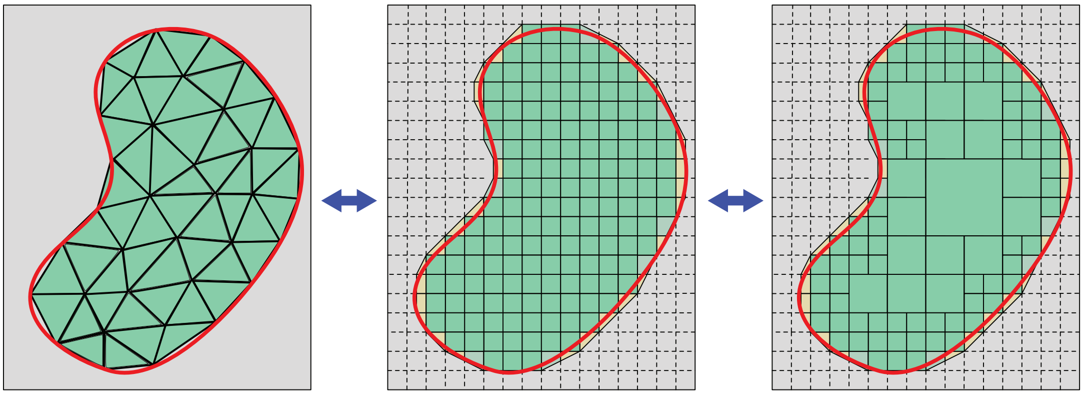


Fig. 1. Unstructured vs. Structured Grids

4 Conclusions

Our current work tries to combine the advantages of both grid types. The goal is to develop data structures which can combine the flexibility and adaptivity of unstructured grids with the performance and efficiency of stencil based methods on structured grids. For this purpose, additional stencils (Fig. 1) are introduced for a better approximation of the domain boundary.

On the low-level side these data structures ideally should be general enough to be adaptable by parameters like cache size, width of SIMD units, memory transfer rates, etc. to the underlying hardware, unifying implementations for CPUs and GPUs as much as possible. On the other hand, they should still contain enough problem-specific high-level information to allow multigrid solving methods.

References

1. Williams, S., Vuduc, R., Oliker, L., Shalf, J., Yelick, K., Demmel, J.: Optimizing sparse matrix-vector multiply on emerging multicore platforms. Parallel Computing (ParCo) **35** (2009) 178–194
2. Bell, N., Garland, M.: Efficient sparse matrix-vector multiplication on CUDA. NVIDIA Technical Report NVR-2008-004, NVIDIA Corporation (2008)
3. Augustin, W., Heuveline, V., Weiss, J.P.: Optimized stencil computation using in-place calculation on modern multicore systems. (2009) 772–784
4. Augustin, W., Heuveline, V., Weiss, J.P.: Convey HC-1 hybrid core computer - the potential of FPGAs in numerical simulation. (2011) 1–8

Goal Oriented Adaptivity for Tropical Cyclones

Martin Baumann*[1], Vincent Heuveline[1], Leonhard Scheck[2], Sarah Jones[2]

[1] Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany

[2] Karlsruhe Institute of Technology, Institut für Meteorologie und Klimaforschung
Wolfgang-Gaede-Weg 1, 76131 Karlsruhe, Germany

Abstract. The development and motion of tropical cyclones is controlled by processes on a wide range of temporal and spatial scales. Goal oriented adaptive methods present a promising way to model such multiscale problems. Such a method identifies and resolves accurately through local refinement only those features that are relevant for a given quantity of interest. We apply an adaptive space-time finite element method to a problem related to tropical cyclone dynamics.

Keywords. adaptive finite element method; goal oriented a posteriori error estimator; space-time discretization; Petrov-Galerkin method

1 Goal oriented error estimator

The main objective for many fluid flow problems is the accurate evaluation of a certain quantity that can be defined by a so-called *goal functional* J. Using the dual weighted residual (DWR) method [1, 2], discretizations for the solution of problems modeled by partial differential equations can be optimized such that the quantity of interest can be approximated accurately using a minimial number of unknowns. The method is based on an a posteriori error estimator that takes sensitivity information with respect to the defined goal functional into account. This sensitivity information is obtained as the solution of the *dual problem*, which is the linearization of the original problem – in this context called *primal problem* – with the linearized goal functional as right-hand side. Using the solution of the dual problem it is possible to compute local error indicators $\eta_i \geq 0$ that represent the contribution of each space-time cell to the total error in J,

$$J(u) - J(u_h) \leq \sum_{i=1\ldots N} \eta_i.$$

Fig. 1 shows the iterative adaptation process: The discrete primal and dual problems are solved for the complete time interval. Then the error contribution of each cell is estimated and the discretization is adapted by refining the spatial mesh or reducing the time step size for cells in which the estimated error is large. This procedure is repeated until the total error has decreased to an acceptable value.

[*] *Corresponding author:* `Martin.Baumann@kit.edu`.

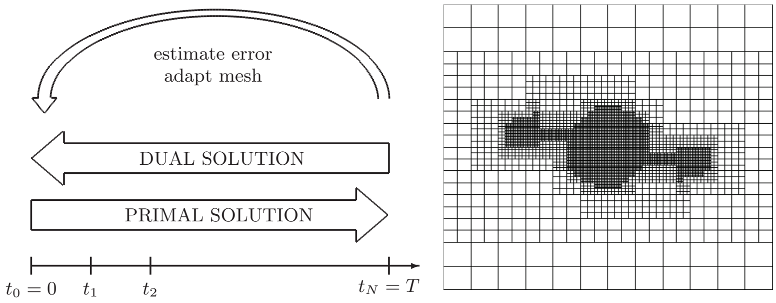


Fig. 1. Left: In each adaption cycle of the DWR method the primal and corresponding dual problems are solved. Right: Optimized mesh for the scenario of interacting storms.

2 Scenario: Two interacting storms

In cooperation with L. Scheck and S. Jones from the Institut für Meteorologie und Klimaforschung (IMK), Karlsruhe Institute of Technology, we applied the adaptive method described in section 1 to an idealized tropical cyclone scenario. In this scenario two cyclones interact, which can lead to complex tracks that depend sensitively on the viscosity parameter and on the temporal and spatial discretization method. Therefore this scenario is an interesting benchmark problem for adaptive methods.

The instationary incompressible Navier-Stokes equations in 2D that have to be solved for this problem are discretized by a space-time finite element method in primitive variables (velocity and pressure). In space the inf-sup stable Taylor-Hood elements [3] and in time the *cGP(1) method* [4] are used. The latter is a Petrov-Galerkin method with piecewise linear trial functions, globally continuous, and piecewise constant test functions that may be discontinuous.

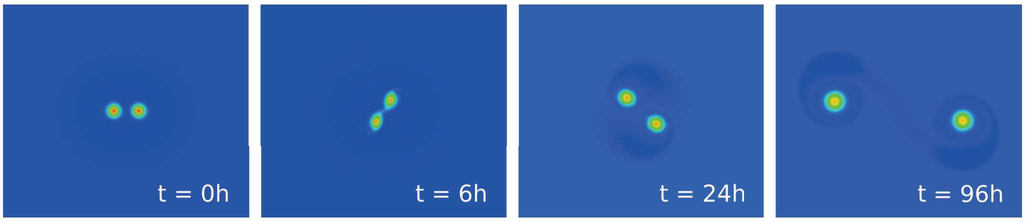


Fig. 2. Storm tracks during the first 96 hours: High vorticity zones (red) indicate the storm positions.

For this scenario, the determination of the storm tracks for the first 96 hours was chosen as the goal of the investigation. We carried out adaptive simulations for several quantitative functionals measuring kinetic energy and vorticity in regions that are related to the cyclone positions. Adaptive runs based on goal functionals that measure vorticity close to the storms, were able to determine the storm positions with high accuracy at low numbers of unknowns of the discrete system. An optimized spatial mesh after two adaption cycles for this scenario is shown in Fig. 1.

Acknowledgements

This work is supported by the Deutsche Forschungsgemeinschaft (DFG-SPP 1276 MetStröm).

References

1. Eriksson, K., Estep, D., Hansbo, P., Johnson, C.: Introduction to Adaptive Methods for Differential Equations. Acta Numerica **4** (1995) 105–158
2. Bangerth, W., Rannacher, R.: Adaptive Finite Element Methods for Differential Equations. Birkhäuser Verlag (2003)
3. Brezzi, F., Falk, R.S.: Stability of Higher-Order Hood-Taylor Methods. SIAM Journal on Numerical Analysis **28** (1991) 581–590
4. Schieweck, F.: A-stable discontinuous Galerkin-Petrov time discretization of higher order. Journal of Numerical Mathematics **18** (2010) 25–57

Validation of Numerical Results in Engineering Applications

Gerd Bohlender, Rudi Klatte, Michael Neaga

Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
bohlender@kit.edu,klatte@kit.edu,neaga@kit.edu

Abstract. In high performance computing, the size and complexity of problems to be solved is growing permanently. At the same time, computing environments are getting more and more heterogeneous. Therefore, there is a growing need for reliability and guaranteed high accuracy of results. This can be achieved by uncertainty quantification methods and consequently numerical verification methods. In this paper, tools for numerical verification are presented and their application on complex engineering problems are discussed.

Keywords. Uncertainty quantification, validated numerical results, interval arithmetic, accurate dot product, C-XSC, HiFlow^3.

1 Introduction

In high performance computing, the size and complexity of problems to be solved is growing permanently. At the same time, computing environments are getting more and more heterogeneous (parallel programming, multithreaded CPUs, GPU programming, FPGA processors, etc.). This means for the user, that the sequence of operations performed is dynamically determined by the system at execution time. In general this makes it impossible to reproduce a computation identically. Therefore, there is a growing need for uncertainty quantification and for reliability and guaranteed high accuracy of results.

In numerical simulation, different sources of error have to be considered: model error (simplification of physical phenomenon), data error (inaccuracy of measurement), method error (numerical approximation method), and rounding error (floating-point arithmetic).

Floating-point arithmetic only delivers approximations of mathematical results. In contrast, interval arithmetic – when correctly applied – always computes an enclosure of the corresponding exact mathematical results even if the sequence of operations is changed. This makes it possible to prove mathematical results in a rigorous way on the computer [12,15]. Different enclosures of the same problem could even be used to improve the accuracy by intersecting the enclosure intervals because the exact solution must be contained in every enclosure.

Dot products make up a large part of all numerical computations. High accuracy may be achieved in many applications by computing dot products exactly or in multiple precision. In an exact dot product, an arbitrary number of products may be accumulated without rounding errors by providing a small number of guard digits. Thus, overflow may be avoided during the lifetime of a computer [13]. An exact dot product avoids cancellation errors and makes the error analysis of numerical methods much easier. Combining interval arithmetic and an exact dot product, verified results with high accuracy can be achieved for many basic numerical algorithms [5,15]. Since 2008 an IEEE standardization group P1788 is working on a standard on interval arithmetic which also includes an exact dot product [8].

Several tools and libraries have been developed for this purpose, e.g. the XSC–languages developed in Karlsruhe [10]. C-XSC is a powerful, free and portable C++ class library for verified scientific computing providing many numerical data types, arithmetic extensions, and other features which support the formulation of algorithms with verified results [9,3,7,4]. INTLAB is another widespread free toolbox for interval arithmetic which is based on the commercial software package MATLAB [14].

2 Performance Issues

In C-XSC (like in all XSC programming languages which have been developed at Karlsruhe University) exact dot products are implemented by means of a so-called *long accumulator*. Using adequate hardware support, exact dot product computations could be made as fast as conventional floating-point approximations. Many algorithms and hardware designs have been developed for this purpose. Unfortunately, in the currently widespread hardware architectures there is no such hardware support. Therefore, the long accumulator has to be simulated in software which makes it considerably slower than floating-point approximations.

Another dot product algorithm named *DotK* is based on highly tuned versions of so-called error free transformations (now commonly called *TwoProd* and *TwoSum*, resp.). k iterations lead to k-fold accuracy of the result (i.e. in contrast to the previous solution, the result is only computed in multiple precision, not exactly). Mathematical properties and implementation details are studied very closely by Rump [15].

In current versions of C-XSC, the user may select different levels of accuracy and performance for each dot product. In addition, the current version of C-XSC provides improved numerical algorithms, BLAS support, multi-threading and better compiler optimizations. Using these new features for the verified solution of a dense linear 1000×1000 system, a speed-up by a factor of nearly 600 could be achieved in comparison with an earlier software version, leading approximately to floating-point speed [4,2,11].

Additional extensions provide support for sparse matrices and parallel programming using MPI.

3 Applications in Engineering Sciences

In [5] the verified solution of many basic numerical algorithms is presented: linear and nonlinear systems, linear and global optimization, automatic differentiation for gradients, Jacobians and Hessians, etc.

In complex applications the numerical treatment normally consists of a sequence of algorithms where the result of one algorithm is the input of the next one. By using verified algorithms, you generally get interval data as the results. So you have to solve an interval problem in the next step which may lead to an inflation of the result. Avoiding this effect requires complex mathematical transformations and the development of completely new algorithms. The additional requirement of automated error control makes the parallelization of such verifying algorithms non-trivial because all parallel processes have to satisfy this condition.

We therefore suppose in a first step to use interval computations and verified algorithms in numerically critical cases. For software development projects a numerical validation component should be available. This requires the following steps:

- Install the current version of C-XSC on a High Performance Environment.
- Execute performance tests using different versions of dot products and different verifying basic algorithms.
- Identify numerically sensible program parts in scenarios solved by means of numerical software tools, e.g. HiFlow^3 [6].
- Apply the C-XSC tools to analyze numerical stability of these program parts.
- Extend this approach to larger modules and problem classes.
- Application of more general tools for uncertainty quantification. This may result in the necessity for developping new models.

4 Conclusion

Uncertainty quantification has to deal with different sources of error: Model errors may be treated by structure and parameter optimization methods, data errors may be handled by rigorous use of intervals or by stochastic methods, method errors may be treated by computation of method error bounds or error estimations. By use of interval methods, method error bounds may be computed automatically. Rounding errors may be minimized by improved basic arithmetic and controlled by using interval arithmetic operations. In this way, numerical results may get a better or even rigorous mathematical quality. Finally, the user has to select the uncertainty quantification method depending on his problem, his requirements of reliability, his available resources, etc.

References

1. G. Bohlender, M. Kolberg, D. Claudio: Improving the Performance of a Verified Linear System Solver Using Optimized Libraries and Parallel Computation. In VECPAR – 8th International Meeting of High Performance Computing for Computational Science, Toulouse. 2008. Lecture Notes in Computer Science, Volume 5336, pp. 13–26, 2008.
2. G. Bohlender, U. Kulisch: Fast and Exact Accumulation of Products - Required by the IEEE Standards Committee P1788. Para 2010 State of the Art in Scientific and Parallel Computing. 10 pages, to appear 2011.
3. W. Hofschuster, W. Krämer: C-XSC 2.0: A C++ Library for Extended Scientific Computing. In Numerical Software with Result Verification. Lecture Notes in Computer Science, Volume 2991, Springer-Verlag, pp. 15–35, 2004.
4. C–XSC version 2.5. Download from `http://www2.math.uni-wuppertal.de/wrswt/` or `http://xsc.de/` Accessed March 10, 2011.
5. R. Hammer, M. Hocks, U. Kulisch, D. Ratz: C++ Toolbox for Verified Computing: Basic Numerical Problems. Springer–Verlag, Berlin / Heidelberg / New York, 1995.
6. HiFlow^3 Documentation and download from `http://www.numhpc.org/hiflow3` Accessed July 24, 2011.
7. W. Hofschuster, W. Krämer, M. Neher: C-XSC and Closely Related Software Packages. Preprint 2008/3, Universität Wuppertal; published in: Dagstuhl Seminar Proceedings 08021 – Numerical Validation in Current Hardware Architectures, Lecture Notes in Computer Science, Volume 5492, Springer-Verlag, pp. 68–102, 2008.
8. IEEE Society IEEE Interval Standard Working Group - P1788. Website: `http://grouper.ieee.org/groups/1788/` Accessed March 10, 2011.
9. R. Klatte, U. Kulisch, C. Lawo, M. Rauch, A. Wiethoff: C–XSC, A C++ Class Library for Extended Scientific Computing. Springer-Verlag, Berlin/Heidelberg/New York, 1993.
10. R. Klatte, U. Kulisch, M. Neaga, D. Ratz, Ch. Ullrich: PASCAL–XSC — Language Reference with Examples. Springer-Verlag, Berlin/Heidelberg/New York, 1992.
11. W. Krämer: High Performance Verified Computing Using C-XSC. Para 2010 State of the Art in Scientific and Parallel Computing. Download from `http://vefir.hi.is/para10/extab/para10-paper-33.pdf` Accessed March 10, 2011.
12. U. Kulisch, R. Lohner, A. Facius (eds.): Perspectives on Enclosure Methods. Springer-Verlag, 2001.
13. U. Kulisch: Computer Arithmetic and Validity Theory, Implementation, and Applications. de Gruyter, 2008.
14. S.M. Rump: INTLAB – INTerval LABoratory. Download from `http://www.ti3.tu-harburg.de/~rump/intlab/` Accessed March 10, 2011.
15. S.M. Rump: Verification methods: Rigorous results using floating-point arithmetic. Acta Numerica, pages 287-449, 2010.

Particle Deposition in the Lungs

Thomas Gengenbach

Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
gengenbach@kit.edu

Abstract. The preparative tasks for a comparison between a patient specific, a schematic and an analytical model for the escape rate of particles in the human lungs are described. It is shown, how to extract specific parameters from a patient specific geometry and build a schematic 3d model with these parameters.

Keywords. particle deposition, lungs, fluid flow simulation, escape rate

1 Introduction

With every breath we take, we are not only breathing in pure air, but also particles like dust that our respiratory tract has to filter and remove from the system. The distribution of particles in the lungs yields an insight, which types of particles are filtered on the way through the respiratory system, and which types make it all the way down to the alveoles. Knowledge of the distribution is helpful to judge the impact of fine dust on the lungs, controlling the efficacy of drugs delivered through the airway system, understanding the defensive mechanisms of the lungs and many more aspects.

The goal of this work is to distinguish between the difference of patient specific and general aspects of particle distributions. To reach this, we start with a three-way comparison of the escape rate. An analytical model was recently presented by Filoche et al. in [1] that holds for different Reynolds numbers and depends primarily on the Stokes number

$$St = \frac{\rho_p d_p^2 u_{fluid}}{18 \mu D}, \tag{1}$$

with the particle density ρ_p, the particle diameter d_p, the characteristic velocity of the underlying fluid field u_{fluid}, the viscosity μ of the fluid and the characteristic system length D parametrized with h, θ and α, whereas h is the ratio of diameters of subsequent bifurcations, α is the angle between bifurcations and θ is the angle between the children tubes of one bifurcation. The second model is a geometrical model that uses exactly the parameters just described extracted from edited CT data to create a three-dimensional geometry of the bronchial tree while the last part of the comparison comprises the geometry of the bronchial tree as segmented from CT scans of the lungs.

This extended abstract gives a short overview of the preparative tasks to start the comparison.

2 Analytical Model

In [1] an analytical model was introduced, which predicts the escape rate of particles using three parameters h, θ and α (cf. Figure 1) and the Stokes number (see Eq. (1)). The escape rate has found to be multiplicative, which basically means that computing the escape rate for one generation and taking the $n-$th power is the same, as computing the escape rate for n generations (cf. Figure 2). The length L to radius $R = d_p/2$ ratio in each bronchiole for the bifurcations is assumed to be $L/D = 3$ and each bifurcation is assumed to be planar (cf. Section 4).

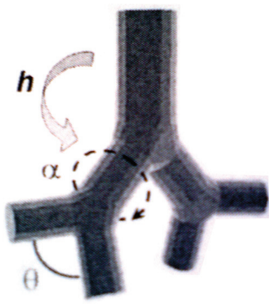


Fig. 1: One bifurcation with the parameters h, θ and α, whereas h is the ratio of diameters of subsequent bifurcations, α is the angle between bifurcations and θ is the angle between the children tubes of one bifurcation.

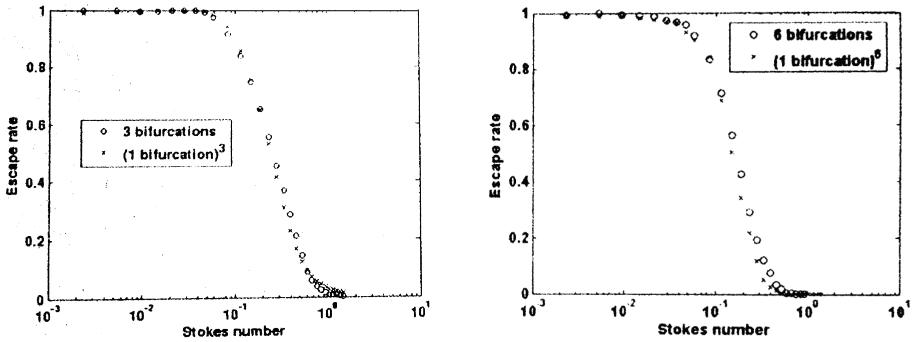


Fig. 2: Plots comparing the results of the escape rate in one bifurcation to the third resp. sixth power and the results of three resp. six simulated generations.

3 Particle Flow

In this first approach, the path of the particles is calculated with a one-way coupling under the presumption that the underlying velocity field was already calculated. Due to the research presented in [1], the Reynolds number of the fluid plays a minor role, hence it can be quite low. The particles are treated separately, solving a transport problem for each particle at each time step. The governing equation is

$$\frac{d}{dt}(m_{particle}v_{particle}) = F_{drag}[+F_{gravity} + F_{?}], \tag{2}$$

with the mass $m_{particle}$ and the velocity $v_{particle}$ of the particles. Forces can be added on the right hand side, but the most important one is the drag force

$$F_{drag} = c * (v_{fluid} - v_{particle})^2, \tag{3}$$

with a constant c that hides the shape and the dynamic viscosity of the particles and the velocities of the fluid minus the velocities of the particles squared. As starting velocity at the entrance of the trachea, the velocity of the underlying fluid is assumed. The resulting motion of the particle in the lungs is

$$\Delta s_{particle} = \Delta t(v_{particle} + v_{fluid}). \tag{4}$$

This corresponds to an Arbitrary Lagrange-Euler (ALE) ansatz.

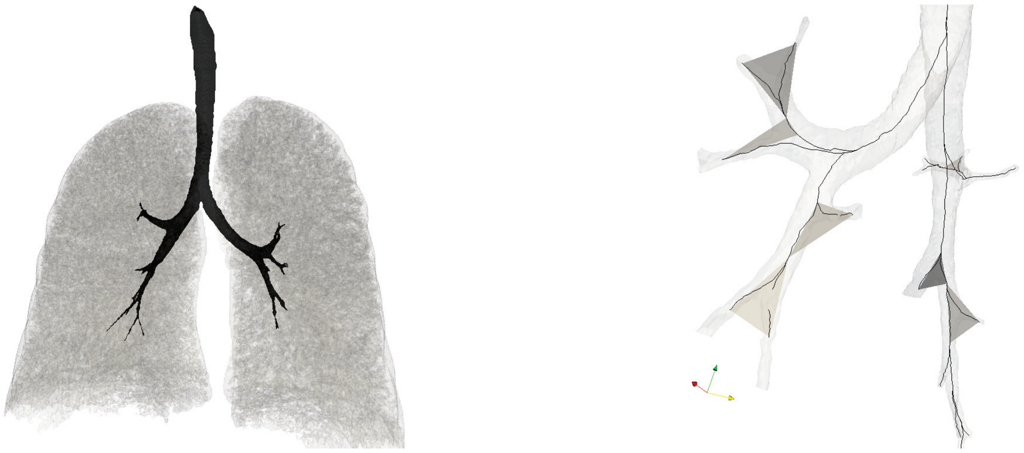
In the upcoming research, we will get rid of this discrete ansatz and treat the particles with a continuous approach, solving the Navier-Stokes equations and a convection-diffusion equation for the particles at each time step, hence the underlying velocity field needs not to be precomputed and the results for the escape rate will not depend on the number of particles at the entrance of the trachea.

4 A Schematic Tree

In the following section, we describe the steps that are necessary to build a schematic tree using the three parameters h, θ and α.

To parametrize the analytical model as well as to create a geometric bifurcation model h, θ and α need to be extracted from the geometry segmented from a CT scan (cf. Figure 3a).

In order to get a real advantage of the analytical model, the extraction process will need to be automatized in the main components. However this is not the focus in this work, hence the parameters were extracted interactively with the use of the Vascular Modeling Toolkit (VMTK) [2]. VMTK is able to extract centerlines (cf. Figure 3b) of lungs and to export the maximal diameter of each bronchiole. Thus it is possible to determine the radius ratio h of subsequent bronchioles. To determine θ and α, we inscribe triangles (cf. Figure 3b) in the geometry using Paraview [3]. Some simple calculations in a table are used to compute the parameters and to verify the assumptions in Section 2 for the bifurcations. A schematic tree build under the use of the extracted parameters is needed to achieve the proposed comparisons. The basic bifurcation, which can be modified (scaled, rotated, etc.), is build similar to the one proposed by



(a) The segmented patient specific geometry (black) of the trachea up to a maximum of seven generations.

(b) Patient specific bronchiole tree with centerlines and inscribed triangles.

Fig. 3: View of the patient specific bronchiole tree in two different editing stages.

Lee et al. in [4], but with less attention to the specifics at the carinal ridge. Several bifurcations are then plugged together by a script to get a bronchiole tree that resembles the patient specific tree with respect to the extracted parameters.

5 Conclusion

With this preparative steps done, we will start to simulate the fluid flow and the particle deposition of the patient specific and the schematic bronchiole tree. We will investigate the differences between the particle escape rate for different Reynolds numbers and particle sizes. Simulations with and without gravity will be done to investigate the impact on the positioning of the patients. After that, the deposition prediction of the analytical model will be compared to the simulated particle depositions.

The following steps comprise an asymmetric schematic tree and instationary fluid flows and particle simulations and further investigations of the analytical model.

References

1. De Vasconcelos, T.F., Sapoval, B., Andrade Jr., J.S., Grotberg, J.B., Filoche, M.: Particle capture into the lung made simple. (2011) To appear.
2. Steinman, D., Antiga, L.: VMTK - Vascular Modeling Toolkit. Webpage (2008)
3. Henderson, A.: ParaView Guide, A Parallel Visualization Application. (2007)
4. Lee, D., Park, S.S., Ban-Weiss, G.A., Fanucchi, M.V., Plopper, C.G., Wexler, A.S.: Bifurcation Model for Characterization of Pulmonary Architecture. The Anatomical Record: Advances in Integrative Anatomy and Evolutionary Biology **291** (2008) 379–389

Towards a multi-purpose modelling, simulation and optimization tool for liquid chromatography in research and teaching

Tobias Hahn

Karlsruhe Institute of Technology (KIT)
Engineering Mathematics and Computing Lab (EMCL)
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
tobias.hahn@kit.edu

Abstract. Simulation and optimization of chromatographic processes are gaining importance especially in industry, where it helps to assure a certain level of quality. Existing software tools are currently limited in their capabilities, such that a new chromatography simulator was designed, built and validated on the basis of experimental data. It satisfies the needs of both, research and teaching.

Keywords. Liquid chromatography, simulation, parameter estimation

1 Introduction

In many areas of biotechnology and bioprocess engineering, chromatography plays a prominent role. Its application in industry ranges from high-resolution analysis of protein mixtures to preparative production of biopharmaceutically active substances. Especially for biopharmaceutical processes it is of utmost importance to guarantee a high level of quality, which cannot be provided on the basis of predictions that rely only on empirical data. Far better insight and more precise predictions can be achieved using mechanistic models that have been developed for chromatographic applications. In academia, those models are already in use for simulations and offline analyses, but still with limited capabilities.

2 Requirements

A scientific software tool for process design and optimization will only be employed if results can be achieved in reasonable time. So far, models assuming homogeneous concentration distribution within a cross section of the chromatography column achieved good agreement with experimental results, meaning a 1D macro scale model is sufficient for most applications [1]. On the other hand, the meso scale models introduce another dimension and often non-linearities. Thus, either a well tuned low-order solver or a sophisticated time stepping scheme allowing larger step sizes has to be employed. As it might require several hundred simulation runs in order to perform optimization, the simulation time of a single chromatography process should be in the range of seconds. Besides the underlying solver, the software should be portable to any operating system.

Another aspect to be considered is data in- and output. Ideally, experimental results should be usable as reference data for e.g. parameter estimation and all generated output data, especially of intra column and intra adsorber processes, should be available for analysis and post processing.

The developments in computer architecture allow more and more fine grained mechanistic models to be used for simulations. On the other hand, the sheer mass of parameters sure is overwhelming for a student, who is new to this topic. In order to investigate for the influence of particular parameters, a simulation software is an ideal tool, as it can predict the outcome of a process that would take e.g. 30 minutes in real life in just a few seconds. While the need for speed is the same as in research, for educational purposes, the output requirements are different: as process design or optimization is not the goal here, only a good representation of the final chromatogram is necessary and an intuitive user interface to facilitate the first steps in this new field. Again a multi-platform approach is inevitable.

3 Existing chromatography tools

To the author's knowledge, only one stand-alone chromatography simulation tool exists, the Chromulator by Gu [2]. This Windows tool is quite versatile but has not been developed further recently. Other simulators built on multipurpose solvers, which of course provide high performance but are not inviting for beginners. There is furthermore a web-based tool built on top of an extensive database with limited computation possibilities [3]. It is easy to use but lacks in flexibility.

4 A new chromatography software

The basis of our new software does not differ widely from the ones mentioned above: A 1D model is discretized with finite elements and solved using fast direct or iterative solvers.

Model

The following processes of concentration change are modelled with homogeneity assumptions for column and beads in 1D (cf. Fig. 1):

1. Convection and diffusion in interstitial phase c_i,
2. Diffusive transport through film layer,
3. Diffusive transport in bead pore phase $c_{p,i}$,
4. Adsorption/desorption in bead phase q_i,

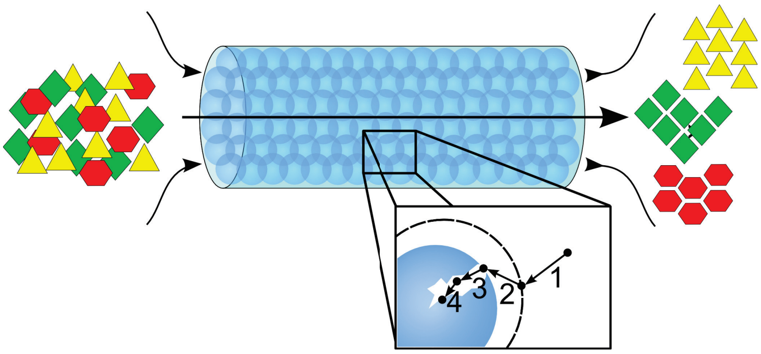


Fig. 1. Modelled processes.

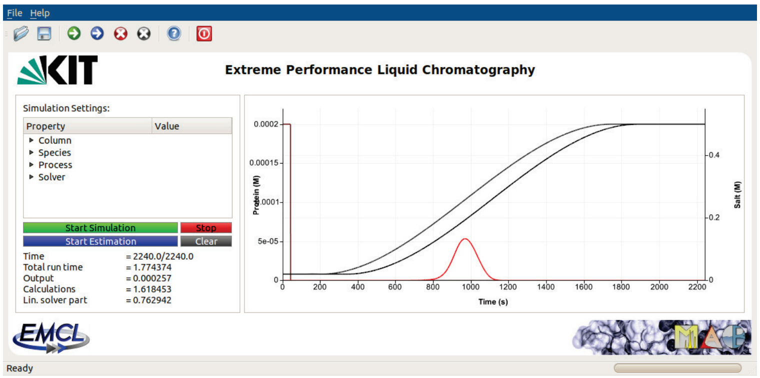
resulting in a system of parabolic and non-linear ordinary differential equations:

$$
\begin{aligned}
\frac{\partial c_i}{\partial t} &= -u\tfrac{\partial c_i}{\partial x} + D_{ax}\tfrac{\partial^2 c_i}{\partial x^2} - k_f\tfrac{3}{r_p}\tfrac{1-\varepsilon_c}{\varepsilon_c}\left(c_i - c_{p,i}\right) && (1+2),\\
\frac{\partial c_{p,i}}{\partial t} &= -\tfrac{1-\varepsilon_p}{\varepsilon_p}\tfrac{\partial q_i}{\partial t} + k_f\tfrac{3}{r_p}\tfrac{1}{\varepsilon_p}\left(c_i - c_{p,i}\right) && (2+3),\\
\frac{\partial q_i}{\partial t} &= k_{a,i}c_{p,i}\left(\Lambda - \textstyle\sum_{j\geq 1}(\nu_j+\sigma_j)q_j\right)^{\nu_i} - k_{d,i}q_i c_{p,0}^{\nu_i} && (4).
\end{aligned}
$$

Symbol	Denotation	Description
u	Fluid velocity	Volume flow per non-solid area
D_{ax}	Axial dispersion	Diffusion and packing deviations
k_f	Film diffusion	Transport to particle surface
r_p	Bead radius	Occurs as $r_p = 3V_{beads}/A_{beads}$
ε_c	Column porosity	V_{fluid}/V_{column}
ε_p	Bead porosity	V_{pores}/V_{beads}
Λ	Ionic capacity	Initial adsorber charge
k_a	Adsorption	Protein adsorption rate
k_d	Desorption	Protein desorption rate
ν	Char. charge	Charge of the adsorbing protein
σ	Shielding	Surface shielding of the protein

Numerical simulation

The stepping scheme used for time-discretization is of second order (Crank-Nicolson) which proved to be a good compromise. First order finite elements provide sufficient accuracy in space. The solvers applied to the resulting linear systems are a GMRES solver, similar to the one used in the LAtoolbox of HiFlow^3 [4]. If available, UMFPACK [5] can be used as well.



Time = 2240.0/2240.0
Total run time = 1.774374
Output = 0.000257
Calculations = 1.618453
Lin. solver part = 0.762942

Fig. 2. The new chromatography software: chromatogram of a gradient elution of a singe protein (red) with smooth salt gradient (black), input values in dark red and gray respectively.

User interface

In order to meet the requirements of both research and teaching, an appealing GUI was designed using Qt [6], that can be compiled for Linux, Windows and other platforms without changes in the code. Simulation settings are imported from XML files and can be edited within the program (Fig. 2).

Chromatograms can be easily analysed in a VTK-powered, interactive chart and post-processed using other VTK-based tools. For example, videos of the intra-column processes can be quickly generated using ParaView [7]. For convenience, a chart of the intra-column view is integrated in the GUI (Fig. 3).

Multi-component experiments can be simulated within seconds, thanks to vectorization and parallelization techniques. Using the integrated plot history, the effects of changed parameter values on the chromatogram can be graphically studied (Fig. 4).

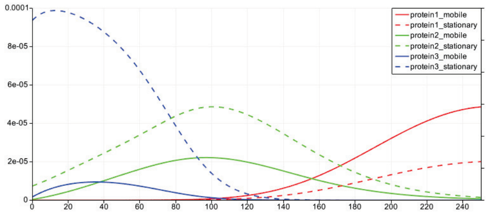


Fig. 3. Plot of intra-column processes

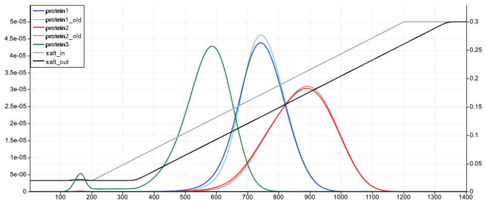


Fig. 4. Simulations of two three-component experiments

Parameter estimation

Estimation of unknown protein parameter values from reference data is supported as well. Here, the whole curve fitting process takes about one minute on a quad-core computer. The underlying method is a Levenberg-Marquardt algorithm with the residual being the Euclidean norm of the difference between measurement and model response. Figures 5 and 6 show a chromatography simulation reconstructed from an experiment.

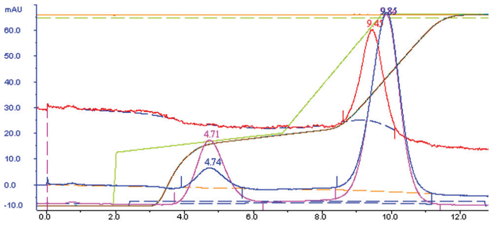


Fig. 5. Measurements from a chromatography experiment

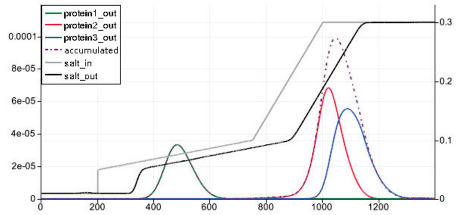


Fig. 6. Simulation from estimated protein parameters

5 Summary and outlook

A new tool for the simulation of liquid chromatographic processes has been designed, built and validated. Compared to other existing tools, it combines the advantages of state of the art models and solvers with an easy to use graphical interface. Further extensions of the software will include the preprocessing of experimentally measured data as well as the possibility to use other types of chromatography besides ion exchange. In addition to this, the possibility of multi-stage processes will be included and special attention will be given to the robustness of estimation and optimization.

References

1. Schmidt-Traub, H.: Preparative Chromatography of Fine Chemicals and Pharmaceutical Agents. Wiley-VCH (2005)
2. Gu, T.: Tingyue Gu's Chromatography Simulation Home Page. http://-oak.cats.ohiou.edu/textasciitilde gu/CHROM (accessed 25.03.2011)
3. Boswell, P., Stoll, D.: hplcsimulator.org – the free, open-source HPLC simulator project. http://hplcsimulator.org (accessed 25.03.2011)
4. Anzt, H., et al.: HiFlow - a flexible and hardware-aware parallel finite element package. EMCL Preprint Series (2010)
5. Davis, T.A.: Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. ACM Transactions on Mathematical Software, vol 30 (2004)
6. Blanchette, J., Summerfield, M.: C++ GUI Programming with Qt 4 (2nd Edition). Prentice Hall (2008)
7. Henderson, A.: ParaView Guide, A Parallel Visualization Application. Kitware Inc. (2007)

Domain Decomposition Method in Optimal Control for Partial Differential Equations

Eva Ketelaer

Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
eva.ketelaer@kit.edu

Abstract. A domain decomposition method (DDM) for optimal control problems constrained by partial differential equations (PDEs) is motivated and derived. Particularly the meaning of applying the Steklov-Poincaré/Schur complement operator and the Neumann-Neumann preconditioner on the continuous and discrete level is shown.

Keywords. domain decomposition method, optimal control for PDEs, Neumann-Neumann preconditioner

1 Motivation

The model problem is an optimal control problem on the domain Ω (Fig. 1) which is constrained by PDEs. The domain Ω is decomposed into subdomains Ω_i. For simplicity it is only divided into two subdomains Ω_1 and Ω_2. It holds: $\partial\Omega \cap \partial\Omega_i \neq \emptyset\,(i = 1, 2)$ (see Fig. 2). The aim of the domain decomposition method is to find an equivalent formulation on the subdomains, which is only coupled through transmission conditions holding on the skeleton $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$ (see Fig. 2).

On the one hand the method is motivated by the need to solve fully coupled large systems. Such systems occur in real world applications and cannot always be solved sequentially due to time or memory restrictions. On the other hand this method implicitly leads to a parallel method. It allows to exploit multi-core

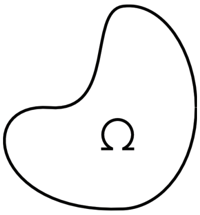


Fig. 1. Domain Ω.

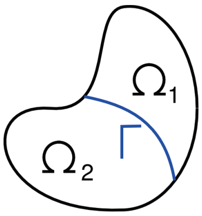


Fig. 2. Subdomains Ω_1, Ω_2 and skeleton Γ.

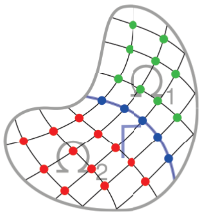


Fig. 3. Discretization of Ω_1, Ω_2 and Γ.

architectures directly, as in any modern laptop or high performance cluster to solve the problem in parallel. Of course, the scalability of the resulting parallel algorithm must be guaranteed.

2 Derivation of DDM on the continuous and discrete level

First the derivation of the equivalent formulation on the continuous level is summarized. For more details see chapter 4 in [1] and [2].

The derivation of the DDM is explained for the following optimality system: Find $y, p \in H_0^1(\Omega)$, $u \in L^2(\Omega)$, such that:

$$\begin{aligned} a(y,\phi) &= \frac{1}{\alpha}(p,\phi)_\Omega + (f,\phi)_\Omega && \forall \phi \in H_0^1(\Omega)\,, \\ a(p,\phi) &= -(y-\hat{y},\phi)_\Omega && \forall \phi \in H_0^1(\Omega)\,, \end{aligned}$$

with a regularization parameter $\alpha > 0$ and $f\,, \hat{y} \in L^2(\Omega)$ given. The corresponding tracking type cost functional is given by

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

The linear form and respectively the bilinear form are defined by:

$$(y,\phi)_\Omega := \int_\Omega y\phi\, dx \text{ and } a(y,\phi) := (\nabla y, \nabla \phi)_\Omega\,.$$

First an equivalent formulation on the two subdomains is derived. This formulation is coupled through transmission conditions of Neumann and Dirichlet type on the skeleton. Then an iterative Neumann-Neumann method, analogously to DDMs for PDEs [3], is used to decouple the formulation:

1. An optimal control problem with Dirichlet boundary is solved on each subdomain in parallel.
2. An optimal control problem with Neumann boundary is solved on each subdomain in parallel.
3. Update of the solution.
4. Go to 1 until convergence.

The update step 3. can be interpreted as a Richardson procedure with the operator $P^{NN} := (\sigma_1 \mathcal{S}_1^{-1} + \sigma_2 \mathcal{S}_2^{-1})$ as a preconditioner. $\mathcal{S}_i$ denotes a Steklov-Poincaré operator in the continuous case. The concrete definitions of the operator can be found in [1]. σ_1, σ_2 are positive weights.

Using the finite element method for the discretization and ordering the degrees of freedom as shown in Fig. 3 we get a linear system of the following type:

$$\begin{bmatrix} B_{11} & 0 & B_{1\Gamma} \\ 0 & B_{22} & B_{2\Gamma} \\ B_{\Gamma 1} & B_{\Gamma 2} & B_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_\Gamma \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_\Gamma \end{bmatrix} .$$

This leads to the Schur complement equation

$$S v_\Gamma = r \,, \tag{1}$$

$$\begin{aligned} \text{with} \quad S &:= B_{\Gamma\Gamma} - B_{\Gamma 1} B_{11}^{-1} B_{1\Gamma} - B_{\Gamma 2} B_{22}^{-1} B_{2\Gamma} \,, \\ r &:= f_\Gamma - B_{\Gamma 1} B_{11}^{-1} f_1 - B_{\Gamma 2} B_{22}^{-1} f_2 \,. \end{aligned}$$

The Schur complement operator S is the discrete counterpart of the Steklov-Poincaré operator.

3 Solution process

Since the Schur complement operator is ill-conditioned, a preconditioned iterative method e.g. the flexible GMRES method [4] is used to solve (1). Only the relevant parts of the flexible GMRES method (Algorithm 1) are printed. We depict particularly the meaning of applying the Steklov-Poincaré/Schur complement operator (line 1 and 4) and respectively the Neumann-Neumann preconditioner operator (line 3) on the continuous level as well as on the discrete level.

Algorithm 1 (Flexible GMRES)

```
1: Compute $Res_0 = r - S(y_\Gamma, p_\Gamma)_0$
2: For $j = 1, \ldots, m$:
3:         Compute $z := P^{NN}(v_\Gamma, q_\Gamma)_j$
4:         Compute $w := S(z)$
5:         [...]
6: If termination criterion is satisfied STOP, else set $(y_\Gamma, p_\Gamma)_0 :=$
   $(y_\Gamma, p_\Gamma)_m$ and go to 1.
```

The matrix G_i $(i = 1, 2)$, needed in the following, is defined as:

$$G_i := \begin{pmatrix} A^i_{\Gamma I} & -\frac{1}{\alpha} M^i_{\Gamma I} & A^i_{\Gamma\Gamma} & -\frac{1}{\alpha} M^i_{\Gamma\Gamma} \\ M^i_{\Gamma I} & A^i_{\Gamma I} & M^i_{\Gamma\Gamma} & A^i_{\Gamma\Gamma} \end{pmatrix} . \tag{2}$$

Applying the Steklov-Poincaré operator on the continuous level means to solve first an optimization problem locally (in parallel) on each subdomain with Dirichlet boundary on the skeleton: Find $y_i, p_i \in V_i$ and $u_i \in L^2(\Omega_i)$ $(i = 1, 2)$, such that:

$$\begin{aligned} a_i(y_i,\phi_i) - \frac{1}{\alpha}(p_i,\phi_i)_i &= 0 && \forall \phi_i \in H_0^1(\Omega_i)\,, \\ y_i &= y_\Gamma && \text{on } \Gamma\,, \\ a_i(p_i,\phi_i) + (y_i,\phi_i)_i &= 0 && \forall \phi_i \in H_0^1(\Omega_i)\,, \\ p_i &= p_\Gamma && \text{on } \Gamma\,, \end{aligned}$$

where

$$\begin{aligned} V_i &:= \{\phi_i \in H^1(\Omega_i),\ \phi_i|_{\partial\Omega\cap\partial\Omega_i} = 0\}, \\ J(y_i,u_i) &:= \frac{1}{2}\int_{\Omega_i} (y_i)^2\ \mathrm{d}x + \frac{\alpha}{2}\int_{\Omega_i}(u_i)^2\ \mathrm{d}x - \int_\Gamma \frac{\partial}{\partial n_i} y_i p_\Gamma\ \mathrm{d}s, \\ (y_i,\phi_i)_i &:= \int_{\Omega_i} y_i\phi_i\,dx \text{ and } a_i(y_i,\phi_i) := (\nabla y_i, \nabla\phi_i)_i. \end{aligned}$$

Then the Steklov-Poincaré operator $\mathcal{S}_i$ is applied to the result for y_i and p_i $(i = 1, 2)$.
At the discrete level first the following linear system has to be solved:

$$\begin{pmatrix} A^i_{II} & -\frac{1}{\alpha}M^i_{II} & A^i_{I\Gamma} & -\frac{1}{\alpha}M^i_{I\Gamma} \\ M^i_{II} & A^i_{II} & M^i_{I\Gamma} & A^i_{I\Gamma} \\ 0 & 0 & I^i_\Gamma & 0 \\ 0 & 0 & 0 & I^i_\Gamma \end{pmatrix} \begin{pmatrix} y^i_I \\ p^i_I \\ y^i_\Gamma \\ p^i_\Gamma \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ y_\Gamma \\ p_\Gamma \end{pmatrix}\,.$$

Then applying the Schur complement operator S_i means multiplying the matrix G_i (2) with the solution vector $(y^i_I,\, p^i_I,\, y^i_\Gamma,\, p^i_\Gamma)^T$.

At the continuous level applying the Neumann-Neumann preconditioner relies first on solving an optimization problem locally (in parallel) on each subdomain with Neumann boundary on the skeleton: Find $y_i, p_i \in V_i$ and $u_i \in L^2(\Omega_i)$ $(i = 1, 2)$, such that:

$$\begin{aligned} a_i(y_i,\phi_i) - \frac{1}{\alpha}(p_i,\phi_i)_i &= 0 && \forall \phi_i \in H_0^1(\Omega_i)\,, \\ \frac{\partial}{\partial n} y_i &= v_\Gamma && \text{on } \Gamma\,, \\ a_i(p_i,\phi_i) + (y_i,\phi_i)_i &= 0 && \forall \phi_i \in H_0^1(\Omega_i)\,, \\ \frac{\partial}{\partial n} p_i &= q_\Gamma && \text{on } \Gamma\,, \end{aligned}$$

whereas $J(y_i,u_i) := \frac{1}{2}\int_{\Omega_i}(y_i)^2\ \mathrm{d}x + \frac{\alpha}{2}\int_{\Omega_i}(u_i)^2\ \mathrm{d}x + \int_\Gamma q_\Gamma y_i\ \mathrm{d}s$. Then the Steklov-Poincaré operator $\mathcal{S}_i$ is applied to the result for y_i and p_i $(i = 1, 2)$.
On the discrete level first the following linear system has to be solved:

$$\begin{pmatrix} A^i_{II} & -\frac{1}{\alpha}M^i_{II} & A^i_{I\Gamma} & -\frac{1}{\alpha}M^i_{I\Gamma} \\ M^i_{II} & A^i_{II} & M^i_{I\Gamma} & A^i_{I\Gamma} \\ A^i_{\Gamma I} & -\frac{1}{\alpha}M^i_{\Gamma I} & A^i_{\Gamma\Gamma} & -\frac{1}{\alpha}M^i_{\Gamma\Gamma} \\ M^i_{\Gamma I} & A^i_{\Gamma I} & M^i_{\Gamma\Gamma} & A^i_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} y^i_I \\ p^i_I \\ y^i_\Gamma \\ p^i_\Gamma \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ v_\Gamma \\ q_\Gamma \end{pmatrix}\,.$$

Then the Schur complement operator S_i is applied to the solution which means multiplying the matrix G_i (2) with the solution vector $(y_I^i, p_I^i, y_\Gamma^i, p_\Gamma^i)^T$.

4 Conclusion and Perspectives

A DDM for optimal control problems constraint by PDEs was outlined. A Steklov-Poincaré/Schur complement operator and a Neumann-Neumann preconditioner on the continuous/discrete level were derived. In future work the DDM will be extended to optimal control problems constraint by non-linear and/or time-depending systems of PDEs. Suitable transmission conditions as well as adequate preconditioners on the skeleton must be derived. Another challenge will be the extension to real world application.

References

1. Nguyen, H.: Domain decomposition methods for linear-quadratic elliptic optimal control problems. PhD thesis, Rice University (2004)
2. Heinkenschloss, M., Nguyen, H.: Neumann–Neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems. SIAM Journal on Scientific Computing **28** (2006) 1001–1028
3. Quarteroni, Alfio ; Valli, A.: Domain decomposition methods for partial differential equations. Repr. edn. Numerical mathematics and scientific computation. Clarendon Press, Oxford [u.a.] (2005)
4. Saad, Y.: Iterative methods for sparse linear systems. 2. ed. edn. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, PA (2003)

OpenLB Progress Report: Towards Automated Preprocessing for Lattice Boltzmann Fluid Flow Simulations in Complex Geometries

Mathias J. Krause

Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
mathias.krause@kit.edu

Abstract. Ongoing research, aiming to obtain a holistic concept for automated preprocessing for numerical simulations of fluid flows in complex geometries, is presented. The approach is based on Lattice Boltzmann methods which are chosen as discretisation strategy in order to simulate Newtonian almost incompressible fluid flows. The focus is brought to those parts of the concept which have already been realised in the framework of the open source library OpenLB. They are illustrated by considering the preprocessing for a patient-specific nasal cavity obtained from CT scans as an example.

Keywords. automated preprocessing, numerical patient-specific fluid flow simulation, Lattice Boltzmann methods, otolaryngology

1 Introduction

Before a numerical simulation can be started, a representation of the discrete geometry together with the corresponding values for the initial and boundary conditions need to be provided. The required data must meet precise requirements which strongly depend on the considered numerical method. Images of the geometry obtained e.g. by *computer tomography* (CT) scanners and certain measurements constitute the raw data. Especially if one considers medical applications, the geometry of interest is often complex and contains small features that cannot be captured by the latest imaging techniques. Therefore, adequate *preprocessing* techniques need to be developed to enable realistic numerical simulations of physical phenomena. The complexity of the geometry makes manual modifications almost impossible. Furthermore, if one considers repeated preparation of data for a numerical simulation, e.g. for patient-individual flow simulations in hospital's daily routine or for technical reasons like making use of adaptive grid refinement strategies, the importance of a high degree of automatisation of the preprocessing becomes obvious.

2 Concept, Realisation and Application

Lattice Boltzmann methods (LBM) (cf. e.g. [1,2,3]) are chosen as the numerical discretisation method in order to solve fluid flow problems. These methods require a voxel mesh with differently marked regions to distinguish different boundary areas. Such meshes need to be generated from the image data obtained e.g. from a CT scanner. The followed overall strategy for the preparation of the data consists of three main steps:

$$\text{3D image data} \rightarrow \text{Surface mesh} \rightarrow \text{Voxel mesh} \rightarrow \text{LB simulation}\ .$$

In the following, the realisation of the strategy is illustrated for an example, namely the preprocessing for a flow simulation in a human nasal cavity. The steps are depicted in Figure 1.

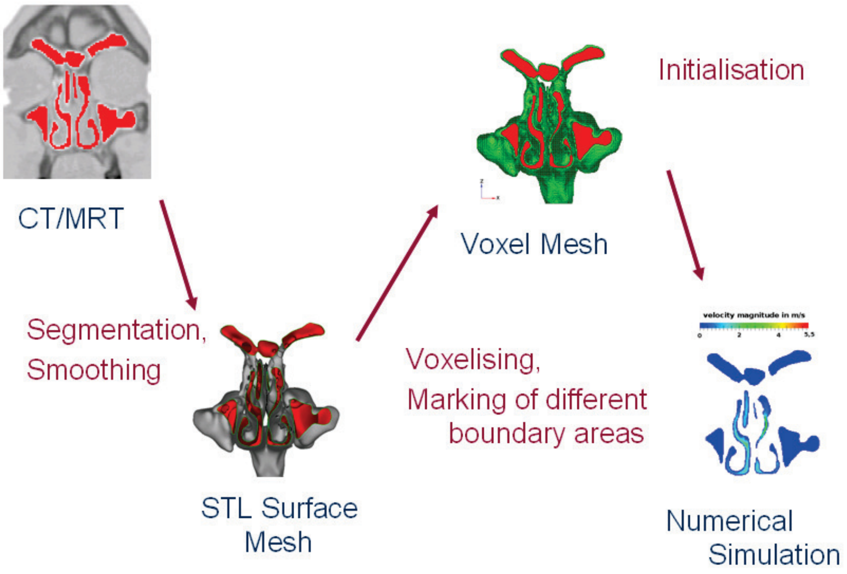


Fig. 1. The preprocessing concept is applied for the preparation of a nasal cavity based on CT data for a LB airflow simulation. The intermediate states are visualised by means of cuts showing the turbinate and paranasal sinuses.

The challenges of the first step, which arise from highly complex geometries that nowadays available CT scanners cannot capture exactly, are faced by taking advantage of *Materialise's*[1] software packages *Mimics* and *3-matics* (cf. [4]). Here, especially the graphical user interface and many partly automated routines enable a high-quality segmentation, followed by a reconstruction of the complete nasal cavity from CT data. For less complex geometries, an automated segmentation and reconstruction of the surfaces is feasible e.g. with open source

[1] http://www.materialise.com

libraries *VMTK*[2] and *ITK*[3]. The volume mesh is obtained fully automatically by applying standard techniques, implemented e.g. in the *CVMLCPP*[4] package. CVMLCPP also offers the computation of the distance from the centre of a boundary voxel to its closest surface which is required for particular interpolation boundary conditions for LBM, e.g. for those proposed by Bouzidi et al. [5]. For the last preprocessing step, an innovative strategy for an automated assignment of different standard boundary conditions dedicated for LBM is applied successfully for complex geometries (cf. [6] and [4]). In the framework of the *OpenLB*[5] project, an interface to CVMLCPP as well as the automated assignment approach has already been implemented. In the near future, the realisation of Bousidi's boundary condition, taking advantage of the CVMLCPP package, and a spatial decomposition of the voxel mesh is planned. This will enable efficient parallel processing within OpenLB (cf. [7]).

References

1. Chopard, B., Droz, M.: Cellular automata modeling of physical systems. Cambridge University Press (1998)
2. Hänel, D.: Molekulare Gasdynamik. Springer (2004)
3. Sukop, M.C., Thorne, D.T.: Lattice Boltzmann modeling. Springer (2006)
4. Krause, M.J.: Fluid Flow Simulation and Optimisation with Lattice Boltzmann Methods on High Performance Computers: Application to the Human Respiratory System. PhD thesis, Karlsruhe Institute of Technology (KIT), Universität Karlsruhe (TH), Kaiserstraße 12, 76131 Karlsruhe, Germany (2010)
5. Bouzidi, M., Firdaouss, M., Lallemand, P.: Momentum transfer of a Boltzmann-lattice fluid with boundaries. Physics of Fluids **13** (2001) 3452–3459
6. Zimny, S.: Numerische Simulation von intranasalen Strömungen mit Lattice-Boltzmann-Methoden. Diplomarbeit, Karlsruhe Institute of Technology (KIT), Universität Karlsruhe (TH), Fakultät für Mathematik (2010)
7. Heuveline, V., Krause, M., Latt, J.: Towards a hybrid parallelization of lattice Boltzmann methods. Computers & Mathematics with Applications (2009)

[2] http://www.vmtk.org

[3] http://www.itk.org

[4] http://tech.unige.ch/cvmlcpp/

[5] http://www.openlb.org

Cloud Computing

Marcel Kunze

Karlsruhe Institute of Technology
Steinbuch Centre for Computing & Engineering Mathematics and Computing Lab
76128 Karlsruhe, Germany
marcel.kunze@kit.edu

Abstract. Cloud computing offers network-centric, scalable, abstracted IT infrastructures, platforms and applications on-demand as utility services. Only the actually consumed resources are subject to accounting and billing. There are three deployment models: Public, private and hybrid cloud. The cloud offerings are delivered as infrastructure, platform or software services. Special services for high performance computing, visualization as well as simulation and optimization could be developed.

Keywords. Cloud Computing, virtualization, services

1 Introduction

Cloud computing is not a new technology, it is a new concept to offer IT services. It may be defined as follows: ”Building on compute and storage virtualization, and leveraging the modern Web, Cloud Computing provides scalable, network-centric, abstracted IT infrastructure, platforms, and applications as on-demand services that are billed by consumption.” [1]

2 Service Deployment Models

There are three deployment models: Public, private and hybrid. In a public cloud, the service provider and the service consumer belong to different organizations. Public clouds usually follow commercial business models and the actual resource usage is being accounted for. The services of a private cloud are always operated by the same organization the consumer belongs to. The motivation for building and running a private cloud may be security and privacy concerns. However, it may be difficult to reach the economy of scale and the availability of a professional public cloud service provider. In a hybrid cloud, services of public clouds and of private clouds are combined. In case of a resource shortfall the public cloud services can be used to satisfy peak loads. Furthermore, it is possible to spread redundant data backups in the cloud to achieve high availability.

3 Service Delivery Models

Infrastructure as a Service (IaaS) allows to operate virtual instances of servers, data stores and networks without the need to physically access the hardware. The clients have full administrative privileges to manage their services and are allowed to define their own level of networking security. IaaS in principle allows to virtualize a complete datacenter and transfer it into the cloud. The most popular public IaaS offering are the Amazon Web Services (AWS). AWS implements for instance compute cycles in the Elastic Compute Cloud (EC2), data storage for Web objects in the Simple Storage Service (S3), and elastic IP adresses.

Platform as a Service (PaaS) is a scalable, integrated application runtime environment and often as well a development platform to support a single or few programming languages. The main target audience are software developers and end users who want to to consume the services in a corresponding market place. A PaaS automatically allows to scale from a single service instance to many. The customer has no need to care about operating system maintenance and installation of application specific software packages. There is almost no administrative overhead in the process of service delivery. A popular public PaaS offering is the Google App Engine.

In a Software as a Service (SaaS) environment complete applications are operated by a provider to be consumed as a utility by the users. In general, no software has to be installed at the local site and the services are available in a Web session. A popular public SaaS offering is the Google Apps environment.

4 HPC as a Service

High Performance Computing as a Service (HPCaaS) is an offer that provides high performance compute resources on-demand over the Internet. Customers of the service are able to provision virtual HPC systems in a self-service portal and deploy and execute their specific application without operator intervention. The business model foresees to only charge the amount of resources actually used. HPCaaS dynamically sizes computing environments appropriate for each individual workload, speeding up the execution of time-critical tasks. However, there remain open questions in the area of performance optimization, advanced resource management, and fault tolerance. The Open Cirrus cloud computing testbed offers an environment in which we can treat these problems.

5 Open Cirrus

Open Cirrus is a cloud computing test bed designed to support research into the design, the provisioning, and the management of cloud services at a global, multi-datacenter scale. Open Cirrus was originally launched by the sponsors HP, Intel and Yahoo! in 2008. In the meantime the testbed is made up by 13 sites in North America, Europe, and Asia. Besides the sponsors there are the following partners: IDA[1], KIT[2], UIUC[3], ETRI[4], MIMOS[5], RAS[6], CESGA[7], CMU[8], CERCS[9] as well as China Mobile/China Telecom. Each site hosts a cluster with up to 1000 cores and associated storage [2].

The open nature of the testbed is designed to encourage research into all aspects of service and datacenter management. Open Cirrus develops an open source cloud stack consisting of physical and virtual machines, and global services such as sign-on, monitoring, storage, and job submission.

The architecture is based on the management of so-called physical resource sets (PRS) at the infrastructure level. These provide logical mini-datacenters to the researchers and isolate the experiments from each other. Based on the physical resource sets it is possible to instantiate virtual resource sets (VRS) in order to get an abstraction from the physical resource layer. The virtualization concept applies to all IT aspects like CPU, storage, networks, and applications. The main advantage of this approach is the potential to create IT services exactly fitting researchers varying needs by automated resource management processes.

6 Conclusions

The concept of cloud computing is currently gaining momentum and yields interesting alternatives to classical IT solutions, both, in enterprise as well as in scientific environments. The concept enables the scientific community to act as a prosumer: We are not only able to easily consume services but also develop and provide new services in the domain of HPC, optimization, and simulation.

References

1. Baun, C., Kunze, M., Nimis, J., Tai, S.: Cloud Computing: Web-basierte dynamische IT-Services. Springer (2009)
2. Avetisyan, A.I., Campbell, R., Gupta, I., Heath, M.T., Ko, S.Y., Ganger, G.R., Kozuch, M.A., O'Hallaron, D., Kunze, M., Kwan, T.T., Lai, K., Lyons, M., Milojicic, D.S., Lee, H.Y., Soh, Y.C., Ming, N.K., Luke, J.Y., Namgoong, H.: Open cirrus: A global cloud computing testbed. Computer **43** (2010) 35–43

[1] Infocom Development Authority, Singapore
[2] Karlsruhe Institute of Technology, Germany
[3] University of Illinois Urbana Champaign, USA
[4] Electronics and Telecommunications Research Institute, South Korea
[5] Malaysian Institute for Microelectronic Systems, Malaysia
[6] Russian Academy of Sciences, Russia
[7] Centro de Supercomputacion Galicia, Spain
[8] Carnegie Mellon University, USA
[9] GeorgiaTech, USA

Highly-Parallel Sweeps for Incomplete LU-factorizations with Fill-ins

Dimitar Lukarski

Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
SRG New Frontiers in High Performance Computing
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
dimitar.lukarski@kit.edu

Keywords. Parallel preconditioners, ILU(p), triangular matrices, multi-core CPU, GPU, multi-coloring, level-scheduling

1 Introduction

Preconditioning techniques are a vital building block for linear system solvers for sparse problems arising from finite element methods (FEM) or related techniques for the solution of partial differential equations (PDEs). High accuracy results in huge systems that are typically sparse, closely coupled, and with bad condition numbers. A typical choice for iterative solvers are Krylov subspace methods like *conjugate gradient* (CG) for symmetric and positive definite systems and the *generalized minimal residual* (GMRES) method. In both cases, the number of iterations depends on the condition number and grows polynomially in the problem size.

2 Parallel Preconditioners

Good preconditioners should fulfill several properties. First, they should mitigate the costs in terms of necessary iterations by restructuring the problem matrix and effecting its spectrum and condition number. Second, since in each iteration step an additional linear system has to be solved, the additional effort should not outweigh achieved benefits. Third – and this point is becoming much more important due to the development towards many-core computing platforms – the preconditioner has to comprise a high degree of parallelism. Fourth, the preconditioner should be applicable to a large class of problems where only minimal additional information is available on specific matrix properties. The latter point is particularly important for the inclusion of preconditioners into widely applicable solver suites. As experience shows, parallelism in preconditioners based on reduced couplings comes at the cost of reduced preconditioning efficiency. So preconditioning also means to find a trade-off between several aspects.

In this work we present our efforts for building highly efficient preconditioning schemes for Krylov subspace iterative solvers in the context of the HiFlow^3 parallel FEM solver package [1]. Our main intention is to provide preconditioners with a high degree of parallelism. Structure and organization of the HiFlow^3

software into several modules and communication layers with unified interfaces for the programmer allows to write a single code base that can be run across a wide range of parallel platforms including multi-core CPUs , graphics processing units (GPUs) and OpenCL-capable accelerators. Since GPU code should be scalable to thousands of threads, our approach is to identify parallelism on the level of blocks within block-decompositions and not only on the level of non-scalable parallel execution of blocks only.

We consider preconditioners in block form based on additive matrix splittings like e.g. Gauss-Seidel and SOR, and multiplicative decompositions like incomplete LU (ILU) decomposition. All our considered preconditioners are based on the block-wise decomposition into small sub-matrices. In both scenarios, typically a large amount of forward and backward sweeps in triangular solvers need to be performed. In order to harness parallelism within each block of the decomposition we use matrix reordering techniques like multi-coloring. For splitting-based methods (Gauss-Seidel, SOR) and ILU(0) without fill-ins the original matrix pattern is preserved. But prior to the solution of the preconditioned system, the matrix is reorganized such that diagonal blocks are diagonal itself. Then, inversion of the diagonal blocks is just an easy vector operation [2]. In the multi-coloring approach sets of independents nodes – called colors – are identified. The number of colors depends on the choice of finite elements and on the dimension. A performance analysis for a convection-diffusion problem solved by Q1 and Q2 elements in two and three dimensions is presented in [5]. The scalability and efficiency of our approach has been demonstrated in [2,3].

Furthermore, we allow fill-ins in the ILU(p) method for achieving a higher level of coupling with increased efficiency. Here we provide two algorithms for parallelism: level-scheduling method [4] and *power(q)-pattern method* [6]. The first method is used as postprocesing method after the factorization to determine the level of parallelism for the elimination processes. Unfortunately, this method produces very small blocks - i.e. the degree of parallelism is very low. An improvement with respect to the level of parallelism of this method can by applied by doing a multi-coloring of the original matrix on first place. As second approach we have developed a new technique. By adding additional colors to the original matrix we can provide a non-zero matrix pattern with diagonal elements in the diagonal blocks after the ILU with fill-ins factorization. Although this algorithm provides only an upper bound for the matrix pattern it provides much higher degree of parallelism compared to the level-scheduling method. We present a detailed performance analysis on multi-core platform and GPU configuration in [6].

3 Preconditioned Solvers in HiFlow^3

The HiFlow^3 finite element software tackles productivity and performance issues by its conceptual approach [1]. Efficient numerical solvers are built on the basis of unified interfaces to different hardware platforms and parallel programming approaches in order to obtain modular and portable software solutions on emerging systems with heterogeneous configuration. With the concept of object-orientation in C++, the HiFlow^3 finite element software provides a flexible, generic and modular framework for building highly efficient parallel solvers and preconditioners for partial differential equations of various types. It comprises a complete set of BLAS 1 and 2 linear algebra routines and data access routines for all platforms. For scalable and portable parallelism all solvers are built upon a two-level library. The linear algebra toolbox (LAtoolbox) orchestrates computations and communications across nodes by an MPI layer. Underneath, the local multi-platform linear algebra toolbox (lmpLAtoolbox) provides highly efficient and optimized implementations with backends to various platforms (e.g. CUDA, OpenMP, OpenCL) where parallelism is expressed by means of data parallelism. Further platforms can be added easily. The user can build modular solutions by utilizing unified interfaces on an abstract level. The same source code for the solver and the preconditioners can be used on several platforms including GPUs. The final choice of platform and particular implementation can be taken at runtime. This approach enables user-friendly and scalable solutions in heterogeneous environments.

4 Conclusion

Combining the above parallel preconditioning techniques and the lmpLAtoolbox we provide portable, flexible and scalable iterative solver combinations with preconditioners based on splitting methods and parallel ILU(p) for many parallel platforms and hardware configurations.

References

1. H. Anzt et al - HiFlow^3 – A Flexible and Hardware-Aware Parallel Finite Element Package *EMCL Preprint Series http://www.emcl.kit.edu/preprints/emcl-preprint-2010-06.pdf* N6 2010
2. V. Heuveline, D. Lukarski, J. P. Weiss - Scalable Multi-Coloring Preconditioning for Multi-core CPUs and GPUs *UCHPC'10, Euro-Par 2010 Parallel Processing Workshops pp.389-297*
3. V. Heuveline, C. Subramanian, D. Lukarski, J. P. Weiss - A Multi-Platform Linear Algebra Toolbox for Finite Element Solvers on Heterogeneous Clusters *PPAAC'10, IEEE Cluster 2010 Workshops*
4. Y. Saad Iterative Methods for Sparse Linear Systems - *Society for Industrial and Applied Mathematics* Philadelphia, PA, USA 2003
5. V. Heuveline, C. Subramanian, D. Lukarski, J. P. Weiss - Parallel Preconditioning and Modular Finite Element Solvers on Hybrid CPU-GPU Systems *The Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering* N36 2011
6. V. Heuveline D. Lukarski, J. P. Weiss - Enhanced Parallel ILU(p)-based Preconditioners for Multi-core CPUs and GPUs – The Power(q)-pattern Method *EMCL Preprint Series http://www.emcl.kit.edu/preprints/emcl-preprint-2011-08.pdf* N8 2011

Parameter estimation and optimal experimental design for the preparative chromatography

Andrea Nestler, Gudrun Thäter

Engineering Mathematics and Computing Lab (EMCL),
Karlsruhe Institute of Technology
76133 Karlsruhe, Fritz-Erler-Str. 23, Germany
Gudrun.Thaeter@kit.edu, Andrea.Nestler@kit.edu

Abstract. Column chromatography is a method for separating and identifying the components of protein mixtures. This work deals with the system of partial and ordinary differential equations describing such separation processes. Furthermore, it is shown how parameter identification can determine properties of single proteins. Optimal results can be achieved by maximizing the correspond sensitivities. The dependence of these sensitivities on the length of the chromatographic column is analysed and interpreted here.

Keywords. preparative chromatography, parameter identification, optimal experimental design

1 Introduction

In column chromatography the so-called *stationary phase* is a solid adsorbent packed in a column. The *mobile phase* [1] is a liquid which is pumped through the packed bed. (see Fig. 1)

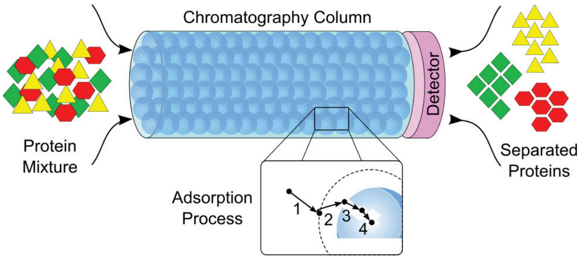


Fig. 1. Column chromatography

The separation of a given protein mixture is achieved in the following way: The stationary phase is an adsorbent - the different proteins adsorb and desorb differently. In addition to the protein mixture a salt concentration is injected into the column. An increasing level of salt concentration elutes the proteins. Hence, by controling the salt concentration different speeds of adsorption and desorbtion are provoked, causing the proteins to leave the column separated from each other.

After separating the proteins the goal is to determine their offen unknown properties, which are condensed in certain parameters which can be estimated by using parameter identification. We are interested in determining these parameters as accurately as possible. Therefore, the associated sensitivities are to be maximized.

2 Model describtion

The process of separating proteins can be described by a system of partial and ordinary differential equations [1]. Let n be the number of different proteins in the mixture. Then the system contains the $n+1$ parabolic differential equations (for each component one convection-diffusion equation)

$$\frac{\partial c_i}{\partial t}(x,t) = \mathbf{D_{ax}} \frac{\partial^2 c_i}{\partial x^2}(x,t) - \mathbf{u_{ax}} \frac{\partial c_i}{\partial x}(x,t) - \boldsymbol{\kappa}_\mathbf{i}[c_i(x,t) - c_{p,i}(x,t)],$$

where $i = 1$ refers to the salt and $i = 2, \ldots, n+1$ to the proteins. Here, $c_i(x,t)$ is the i-th component in the mobile phase and $c_{p,i}(x,t)$ is the concentration of the i-th component in the particle pores.

The boundary conditions are

$$c_i(0,t) = c_{in,i}(t) + \frac{\mathbf{D_{ax}}}{\mathbf{u_{ax}}} \frac{\partial c_i}{\partial x}(0,t) \quad \text{and} \quad \frac{\partial c_i}{\partial x}(L_c,t) = 0\,,$$

and the initial conditions are $c_i(x,0) = 0\,, \forall x \in \Omega$.

For $c_{p,i}(x,t)$ the following $n+1$ ordinary differential equations model a film transfer as

$$\frac{\partial c_{p,i}}{\partial t}(x,t) = \boldsymbol{\eta}_i[c_i(x,t) - c_{p,i}(x,t)] - \frac{1-\boldsymbol{\varepsilon}_\mathbf{p}}{\boldsymbol{\varepsilon}_\mathbf{p}} \frac{\partial q_i}{\partial t}(x,t),$$

where $q_i(x,t)$ is the concentration of the i-th component in the stationary phase. These $2n+2$ equations are called the *lumped rate model.*

Here, $\kappa_\mathbf{i} := 3\frac{k_{eff,i}}{r_p}\frac{(1-\varepsilon)}{\varepsilon}$ and $\eta_\mathbf{i} := 3\frac{k_{eff,i}}{r_p}\frac{1}{\varepsilon_p}$. The following table explaines the constants [1]

$\mathbf{D_{ax}}$	axial dispersion coefficient
$\mathbf{u_{ax}}$	interstitial velocity in the packed column
$\mathbf{k_{eff,i}}$	effective mass transfer coefficient
$\mathbf{r_p}$	particle radius
ε	void fraction
ε_p	porosity of the solid phase

In addition, the following $n+1$ ordinary differential equations describe the so-called SMA model (Steric Mass Action) for ion-exhange chromatography [2]

$$\frac{\partial q_i}{\partial t}(x,t) = \mathbf{k_{a,i}}\, c_{p,i}(x,t)\overline{q}_1(x,t)^{\boldsymbol{\nu}_i} - \mathbf{k_{d,i}}\, q_i(x,t)c_{p,1}(x,t)^{\boldsymbol{\nu}_i}, \quad (i=2,\ldots,n+1)$$

where $\overline{q}_1(x,t) = \boldsymbol{\Lambda}_i - \sum_{i=2}^{n+1}(\boldsymbol{\nu}_i + \boldsymbol{\gamma}_i)q_i(x,t)$ and $q_1(x,t) = \boldsymbol{\Lambda}_i - \boldsymbol{\nu}_i \sum_{i=2}^{n+1} q_i(x,t)$. Hence we have

$$\frac{\partial q_1}{\partial t}(x,t) = -\boldsymbol{\nu}_i \sum_{i=2}^{n+1} \frac{\partial q_i}{\partial t}(x,t).$$

The SMA parameters $\mathbf{k_{a,i}}$ (adsorption coefficient), $\mathbf{k_{d,i}}$ (desorption coefficient), $\boldsymbol{\gamma}_i$ (steric factor) und $\boldsymbol{\nu}_i$ (characteristic charge) should be determined.

3 Parameter estimation

In column chromatography the concentration of the proteins are detected at the end of the column (at $x = L_c$). Now we assume that our experiment was carried out with the result that the different components have left the column at different times. Let $z(t)$ denote the measured data.

The numerical solution of the system of parabolic and ordinary differential equations depends on the SMA parameters. Thus, the concentration of the i-th component in the *mobile phase* at $x = L_c$ has the form

$$u_i(t;\theta_i) := u_i(t;k_{a_i},k_{d_i},\gamma_i,\nu_i) = c_i(L_c,t;k_{a_i},k_{d_i},\gamma_i,\nu_I).$$

We set

$$y(t;\theta_2,\ldots,\theta_{n+1}) := \sum_{i=2}^{n+1} u_i(t;\theta_i).$$

By solving the following optimization problem [3] the $4n$ SMA parameters can be determined

$$\boldsymbol{\theta} := \arg\min_{\boldsymbol{\theta}} \int_T (y(t;\,\boldsymbol{\theta}) - z(t))^2 dt$$

such that $A_i \begin{pmatrix} c_i(\cdot;\boldsymbol{\theta}) \\ c_{p,i}(\cdot;\boldsymbol{\theta}) \\ q_i(\cdot;\boldsymbol{\theta}) \end{pmatrix} = b_i,\ (i=1,\ldots,n+1).$

4 Optimal experimental design

For solving the optimization problem as accurately as possible we try to "maximize" the sensitivities with respect to the parameters. It can be shown that the so-called *Fisher Information Matrix* (FIM)

$$M = \frac{1}{\sigma^2}\int_0^{t_f} \left(\tfrac{\partial y(t;\theta)}{\partial \theta}\right)\left(\tfrac{\partial y(t;\theta)}{\partial \theta}\right)^T dt \to \max$$

is equivalent to the inverse of the covariance matrix [3].
(With $\varepsilon(t;\theta^*) = y(t;\,\theta^*) - z(t)$ we have $\mathbf{E}(\varepsilon(t;\theta^*)) = 0$ and $\mathbf{E}(\varepsilon(t;\theta^*)\varepsilon(t';\theta^*))$ $= \sigma^2\delta(t-t')$. This means that ε is a white Gaussian noise process and $\sigma > 0$ denotes the standard deviation of the measurement noise.)
The maximization of the FIM with respect to an optimality criterion (e.g. maximization of the determinant) is equivalent to "minimizing" the covariance matrix.
In the following, the sensitivities $\frac{\partial y(t;\theta)}{\partial\theta}$ are determined by solving the so-called *sensitivity equations*

$$\frac{\partial}{\partial\theta} A_i \begin{pmatrix} c_i(\cdot;\theta) \\ c_{p,i}(\cdot;\theta) \\ q_i(\cdot;\theta) \end{pmatrix} = \tilde{A}_i \begin{pmatrix} \frac{\partial c_i}{\partial\theta}(\cdot;\theta) \\ \frac{\partial c_{p,i}}{\partial\theta}(\cdot;\theta) \\ \frac{\partial q_i}{\partial\theta}(\cdot;\theta) \end{pmatrix} = 0,\ (i = 1,\ldots,n+1).$$

5 Example

The goal is to determine $det(M)$ in dependence of the length of the column L_c if the protein mixture consists only of one component. In this case we have $M \in \mathbb{R}^{4\times4}$.
We consider a chromatography process with the following input parameters:

$$D_{ax} := 1.574\cdot10^{-4},\ u_{ax} := 0.42,\ \kappa_1 := \kappa_2 := \frac{2}{3},\ \eta_1 := \eta_2 := \frac{10}{3},\ \varepsilon_p := 0.3,$$

$$c_{in,1}(t) := 0.02 \quad \text{and} \quad c_{in,2}(t) := 2\cdot10^{(-4)} \quad \forall\, t \in (0, 1000].$$

The standard deviation is denoted by $\sigma := 5\cdot10^{-6}$.
First we make the simplified assumption that we are only interested in one parameter (e.g. $k_a \in \mathbb{R}$, the other three SMA parameters are assumed to be known). This means that $M \in \mathbb{R}^+$. As we can see in Figure 2 the Fisher Information increases with the length of the column.
Now we assume that we have to determine all four SMA parameters. (none of them is known) In this case it holds that $M \in \mathbb{R}^{4\times4}$ (as mentioned above). Figure 3 shows the dependence between $det(M)$ and the column length L_c.
The maximum of the determinant is achieved at $L_c = 41.5$ mm. Considering the covariance matrix M^{-1} we observe that the non-diagonal elements become larger as the length of the column increases. Therefore, the dependence of the parameters on each other is growing. We assume that $det(M)$ converges to a lower bound, which has to be determined.

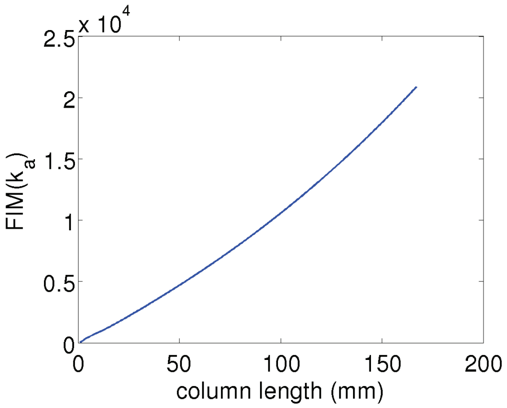


Fig. 2. The Fisher Information of k_a in dependence on the column length. If the exact values of k_d, γ and ν are known we see that the longer the column is, the more sensitive are the measured values in relation to the determination of k_a.

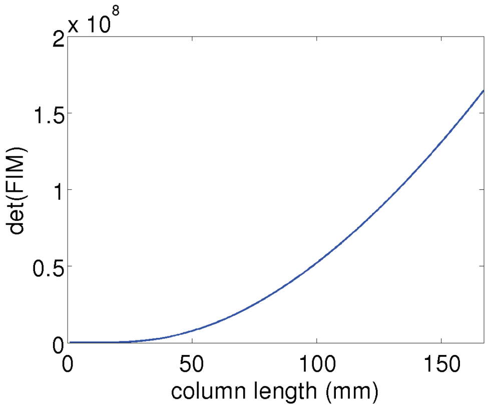


Fig. 3. The determinant of the Fisher Information matrix $det(M(k_a, k_d, \sigma, \nu)$ in dependence on the column length. The longer the column is, the more sensitive are the measured values in relation to the determination of the FIM. It was shown that the dependence of the parameters on each other is growing with a growing length column. That is the reason for the decreasing determinant.

6 Summary and outlook

In section 3, we have briefly shown how to obtain the unknown SMA parameters by *parameter identification*. By using this *parameter identification* technique, we can find a continuous solution to the given optimization problem. Nevertheless, an efficient method has to be developed. Furthermore, it has to be determined how many experiments must be conducted for getting enough information to solve the optimization problem. Further studies will deal with the solvability of the optimization problem in dependence of the number of experiments.
In section 4, the derivation of the so-called sensitivities was shown. We will continue by considering the associated parabolic and ordinary diffenrential equations more precisely with respect to these sensitivities. Such equations should be solved by the finite element method. Particulary the functional analysis aspects have to be taken into account. For example, we are going to study the functional spaces of the solutions.
In section 5, a column chromatography process was described. Some experiments were conducted and the results were shown. Such experiment should be performed for another processes, too. Here, the aim is to determine a general prediction of the optimal column length of a chromatography process.

References

1. Schmidt-Traub, H.: Preparative Chromatography. Wiley Verlag (2005)
2. Brooks, C.A., Cramer, S.M.: Steric Mass-Action Ion Exchange: Displacement Profiles and Induced Salt Gradients. SIAM Journal on Numerical Analysis **38** (1992)
3. Ucinski, D.: Optimal Measurement Methods for Distributed Parameter System Identification. CRC Press (2004)

Mobile Simulation – from Compute-Clouds to Smartphones and Tablets

Sebastian Ritterbusch[1]

Karlsruhe Institute of Technology (KIT)
Engineering Mathematics and Computing Lab (EMCL)
Fritz-Erler-Str. 23, Geb. 01.86, 76133 Karlsruhe, Germany
sebastian.ritterbusch@kit.edu

Keywords. Visualization, Cloud-Computing, Mobile Simulation

Abstract The development of scientific computing including numerical simulation and interactive 3D visualization has today become an essential tool in many applications ranging from industrial design, environmental research and meteorology to medical engineering. At present, the set-up of suitable IT-resources, use of simulation software, and the visualization of the results require usually a dedicated expertise and significant resources. The use of smartphones and tablets to access service-oriented visualization structures is laying a new path to simulation analysis.

1 Introduction

The computational demands for interactive visualization and evaluation of large numerical simulations have been growing steadily, leading to large, expensive and stationary installations. This is is contrast to modern flexible workflows and completely prevents access of the general public of a first-hand impression of visualization results. This is in strong contradiction to the fact, that interactive 3D visualization is more understandable, realistic and can be of high relevance to the general public, more than ever. Mobile access to numerical simulations for scientists, decision makers and the public is desirable for many reasons, ranging from various applications to new interaction and exploration models. This article discusses technical aspects of providing mobile access to numerical simulations, the potential of the novel man-machine interface of current mobile devices, leading to provide visualization as a service from cloud resources.

2 Simulations on Mobile Devices

The two exemplary application areas in medical engineering and urban simulations demonstrate the advantages of providing simulation results on smartphones and tablet computers, illustrated in Figure 1.

In medical imaging, data sets from various devices such as the computer tomograph, x-ray, magnetic resonance tomograph and others are transformed into images using mathematical models. Such data sets are often quite large,

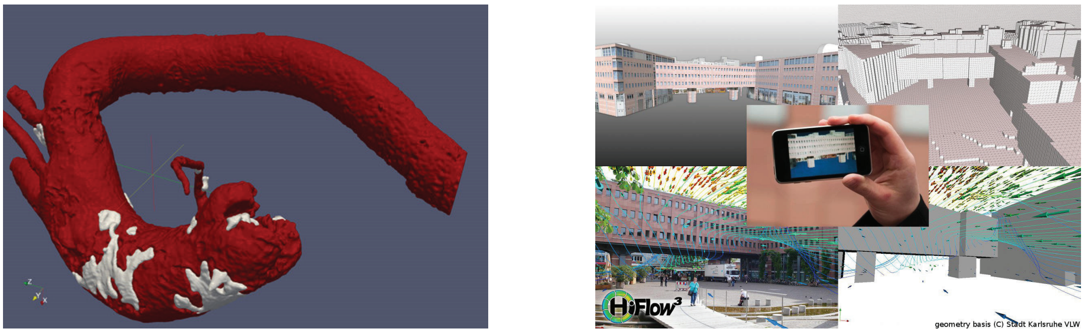


Figure 1. Medical imaging of an aortic valve for operation planning, and the work-flow for urban wind simulation and visualization

and are needed for analysis, operation planning, discussion, illustration and as intra-operational information basis. Furthermore, current simulation technology may enhance the three-dimensional measurements by simulations for example of blood or air flow in the measured body [7,2]. While the data management and design of appropriate computational facilities needs specific attention [9], flexible and immediate access to the results at various places such as the office, hospital room and during the surgery is highly desirable.

Numerical simulations in the world we live in, is made possible by establishing the work-flow for pre-processing urban planning models, numerical simulation and augmented reality visualization, for example in the collaboration of the city council of Karlsruhe with the Engineering Mathematics and Computing Lab (EMCL) going beyond traditional concepts of urban computing [4]. A typical application is urban air flow simulation in current city scapes, which also presents results of scientific computation in a more comprehensible form for citizens by the innovative use of augmented reality visualization. Access to such simulations can be realized as on-demand services by automated geometry pre-processing and localized simulation visualization for augmented reality applications on mobile devices. While such concepts have been analyzed and realized in laboratory setups [5], the availability right on the spot is a new dimension for the usability and availability of such technology.

3 Human Machine Interface

Ease of use was an fundamental aspect of the success of modern smartphones. Computational power was made available to the user at the tip of the finger, based on a rich sensor equipment on modern mobile devices: Ranging from buttons, touchscreen, accelerators, gyroscopes, magnetometers, GPS, distance sensors to camera systems used in combination, novel user interfaces have changed the way how we interact with mobile devices [13]. The appropriate use of the sensor equipment is changing the way, how simulation results can be accessed; for example, the accurate computation of the 6D viewpoint consisting of 3D position and 3D camera orientation, is the starting point to embedding the simulation results into the camera image. The lack of traditional interaction devices, such as mouse and keyboard, is overcome by intuitive spatial interaction concepts, made possible by the sensor equipment.

4 Mobile Device Performance

Although the performance and capabilities of mobile devices have significantly increased, the main design goals are power consumption and cost [6]. The comparison of current HPC simulations illustrates this gap, for example the resources needed for climate research [1] with simulations on systems of theoretical Linpack performance of up to 158 TFlops, are multitudes of magnitudes beyond mobile processors such as the Atom N330 at 1.6Ghz and the ARM Cortex A8 at 0.6GHz were benchmarked at 933MFlops and 23MFlops [12]. Despite the much higher power efficiency, and use of dedicated floating point units are very promising even leading to highly efficient HPC installations, in mobile computing the sparse energy resources are very limiting for computing simulations on the devices themselves.

5 Remote Visualization and Network Reality

A feasible solution to provide the results of numerical simulations on mobile devices is given by remote visualization concepts. Traditionally, remote visualization is based on the frame-by-frame basis. An analysis for the needed bandwidth shows this approach to be problematic for cellular networks: At a resolution of 960×640 RGB pixels and 25 frames per second, an MJPEG codec would need about 30 Mbit/s (at Q:75), a modern and temporal-aware MPEG4 codec can be estimated at 3.75 MBit/s (at 0.25 bits per pixel and frame). The theoretical bandwidth of EDGE cellular networks are at 0.2MBit/s up to 14.4MBit/s for HSDPA UMTS networks, with practical averages at 0.1MBit/s up to 1.2MBit/s, following [10,11]. Even worse, the round-trip latency without pay-load is ranging from 150ms (UMTS) to 600ms (EDGE), and is therefore very noticeable. The Long Term Evolution (LTE) network has a theoretical peak of 100MBit/s and a latency of below 10ms, but may only be available in selected areas. Therefore, more advanced approaches for remote visualization are advisable, such as model reductions and spatial-aware compression methods. By this, high frame-rates are feasible, coupled with low bandwidth usage and de-coupling from current network conditions.

6 Visualization as a Service

The resources needed for visualization and simulation are significant. They are needed dynamically, depending on time of access, amount of data, computational demand, the number of users, and other issues. It is a fundamental aspect to define an efficient framework, for the visualization service, the simulation resources and data management satisfying the need for security and reliability backed by a business model. The implementation as a cloud service [3,8] may offer the opportunity to provide the resources as needed in a secure and reliable fashion, with high data locality as illustrated in Figure 2.

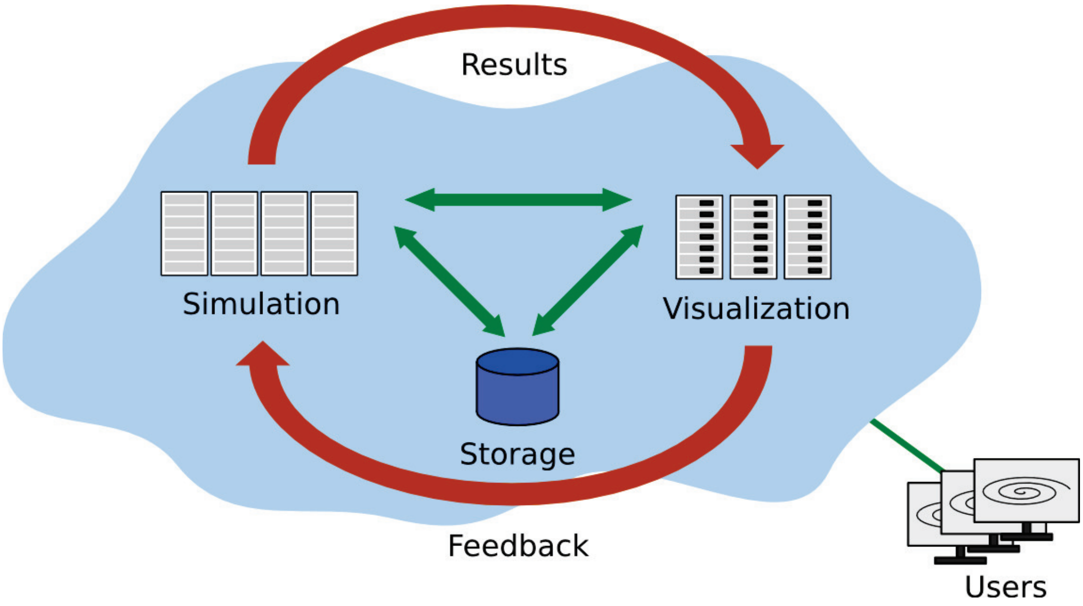


Figure 2. Visualization as a Service

References

1. Panagiotis Adamidis, Vincent Heuveline, and Florian Wilhelm. A High-Efficient Scalable Solver for the Global Ocean/Sea-Ice Model MPIOM. EMCL Preprint Series, 2011.
2. H. Anzt, W. Augustin, M. Baumann, H. Bockelmann, T. Gengenbach, T. Hahn, V. Heuveline, E. Ketelaer, D. Lukarski, A. Otzen, S. Ritterbusch, B. Rocker, S. Ronnas, M. Schick, C. Subramanian, J.-P. Weiss, and F. Wilhelm. HiFlow3 – A Flexible and Hardware-Aware Parallel Finite Element Package. EMCL Preprint Series, 2010.
3. Christian Baun, Marcel Kunze, Jens Nimis, and Stefan Tai. *Cloud Computing: Web-basierte dynamische IT-Services*. Springer-Verlag, 2010.
4. Paul Dourish, Ken Anderson, and Dawn Nafus. Cultural mobilities: Diversity and agency in urban computing. In Cécilia Baranauskas, Philippe Palanque, Julio Abascal, and Simone Barbosa, editors, *Human-Computer Interaction – INTERACT 2007*, volume 4663 of *Lecture Notes in Computer Science*, pages 100–113. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-74800-7_8.
5. Holger Graf, Pedro Santos, and André Stork. Augmented reality framework supporting conceptual urban planning and enhancing the awareness for environmental impact. In *Proceedings of the 2010 Spring Simulation Multiconference*, SpringSim '10, pages 181:1–181:8, New York, NY, USA, 2010. ACM.

6. Farinaz Koushanfar, Vandana Prabhu, Ana Prabhu, Miodrag Potkonjak, and Jan M. Rabaey. Processors for mobile applications. In *International Conference on Computer Design*, pages 603–8, 2000.
7. Mathias Joachim Krause. *Fluid Flow Simulation and Optimisation with Lattice Boltzmann Methods on High Performance Computers - Application to the Human Respiratory System* . PhD thesis, University of Karlsruhe (TH), 2010.
8. Alexander Lenk, Markus Klems, Jens Nimis, Stefan Tai, and Thomas Sandholm. What's inside the cloud? an architectural map of the cloud landscape. In *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, CLOUD '09, pages 23–31, Washington, DC, USA, 2009. IEEE Computer Society.
9. M. T. Nkosi and F. Mekuria. Cloud computing for enhanced mobile health applications. In *CloudCom*, pages 629–633. IEEE, 2010.
10. AT&T Developer Program. Gsm: Umts/hsdpa/hsupa/hspa. online documentation.
11. AT&T Developer Program. Long-term evolution (lte). online documentation.
12. Katie Roberts-Hoffman and Pawankumar Hegde. ARM Cortex-A8 vs. Intel Atom: Architectural and Benchmark Comparisons, 2009.
13. Dongsong Zhang, Boonlit Adipat, and Yaser Mowafi. User-centered context-aware mobile applications – the next generation of personal mobile computing. *Communications of the Association for Information Systems*, 24, 2009.

Elliptic Operators and Finite Element Methods: Error Bounds for Multi-Precision Approaches

Björn Rocker

Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
bjoern.rocker@kit.edu

Keywords. Finite Element Methods, Error Estimation, Mixed Precision, Iterative Refinement Method, Hardware-aware Computing

1 Introduction

For elliptic partial differential equations, there exist the lemmas of Strang and Aubin-Nitsche, providing error estimations when discretizing the operator with finite elements. These estimations can be used to derive an upper bound for the perturbations be brought in during the solution process.

2 Basics: Mixed Precision Iterative Refinement

The Iterative Refinement method (IR) can be interpreted as an inexact Newton method which is applied to the function $f(x) = b - Ax$ with $\nabla f(x) = A$, where $Ax = b$ is the linear system that should be solved:

$$x_{i+1} = x_i - (\nabla f(x_i))^{-1} f(x_i) = x_i - A^{-1} \underbrace{(b - Ax_i)}_{=:r_i}.$$

In each iteration step, the error correction equation $Ac_i = r_i$ is solved approximately based on an arbitrary linear solver.

One promising approach to employ mixed precision techniques is to update the solution approximation in high precision (e.g. double precision) and compute the error correction term c_i in lower precision (e.g. single precision). This approach was suggested and evaluated e.g. in [1], [2] and [3]. It is also possible to perform a nested IR method where the error correction equation is solved again with IR. From the analytical point of view, the nested approach can be interpreted as a variation of a multigrid method.

3 Discretization and Method Error for Elliptic Operators

The task is to search $u \in V$ based on a variational problem of the form

$$a(u, v) = \langle f, v \rangle \quad \forall v \in V, \tag{1}$$

where $V \subset H^m(\Omega)$ and $a : V \times V \to \mathbb{R}$ is a symmetric, elliptic and positive bilinear form. $\Omega \subset \mathbb{R}$ is assumed to be bounded and $f \in L^2(\Omega)$. In the following, $\tilde{a}_h$ describes the representation of a in a certain floating point format on a discretization grid with grid-size h.

When using the mixed precision approach within IR to solve the arising linear system from equation (1), we have to be aware of the fact that transforming the matrix and right hand side into a lower floating point format means to solve a perturbed system describing no longer exactly the problem that shall be solved. Suppose the perturbation reads:

$$\tilde{\tilde{a}}_h(\cdot,\cdot) := \tilde{a}_h(\cdot,\cdot) + b_1. \tag{2}$$

One established strategy to bound the error arising in the solution process is to keep it at maximum in the order of the discretization error. For elliptic operators, the latter can be estimated using the lemma of Aubin-Nitsche. By extending the lemma of Strang it is possible to consider the perturbation within the solving process. By combining both lemmas and statement (2), we can derive an equation that can be used to find upper bounds for the perturbations b_i of a, resp. f:

$$\left(\underbrace{\|u - v_h\|_m}_{\in O(h^2)} + \underbrace{\sup_{w_h \in V_h} \frac{|a(v_h, w_h) - \tilde{a}_h(v_h, w_h)|}{\|w_h\|_m}}_{\in O(h^2)} \right) + \underbrace{\sup_{w_h \in V_h} \left(\frac{|\langle f, w_h\rangle - \langle \tilde{f}_h, w_h\rangle|}{\|w_h\|_m} \right)}_{\in O(h^2)}$$

$$+ \underbrace{\sup_{w_h \in V_h} \left(\frac{|b_1|}{\|w_h\|_m} \right)}_{\overset{!}{\in} O(h^2)} + \underbrace{\sup_{w_h \in V_h} \left(\frac{|b_2|}{\|w_h\|_m} \right)}_{\overset{!}{\in} O(h^2)} \leq \frac{c_a}{c_s} h^2 \|f\|_0. \tag{3}$$

Herby c_a denotes the constant arising in the lemma of Aubin-Nitsche, c_s the one arising in the lemma of Strang.

4 Conclusions

Based on the shown results it might be possible to determine a priori upper bounds for the perturbations of the linear system within the Iterative Refinement method. Thus, the floating point (or arbitrary number-) format within the solution process can be chosen in order to develop highly efficient solvers.

References

1. Buttari, A., Dongarra, J., Langou, J., Langou, J., Luszcek, P., Kurzak, J.: Mixed precision iterative refinement techniques for the solution of dense linear systems. (2007)
2. Anzt, H., Heuveline, V., Rocker, B.: An error correction solver for linear systems: Evaluation of mixed precision implementations. (2011)
3. Göddeke, D., Strzodka, R., Turek, S.: Performance and accuracy of hardware–oriented native–, emulated– and mixed–precision solvers in FEM simulations. (2007)

Numerical Simulation of Microfluidic Optical Components based on Liquid-Liquid Interfaces

Staffan Ronnås

Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
staffan.ronnas@kit.edu

Abstract. A methodology for the numerical simulation of dynamically reconfigurable microfluidic optical components is outlined. The main challenge lies in the tracking of liquid-liquid interfaces during the computation, as well as the interaction between the evolution of the interface and the fluid flow process. Among the existing methods in the literature, the level set method seems to be the most appropriate in that context due to its robustness and flexibility.

Keywords. optofluidics, interface tracking, level set method

1 Optofluidics

Optofluidics is a scientific domain that integrates technologies from two separate areas: microfluidics and optics, with the aim of creating efficient and adaptable devices for use in chemical and biological analysis, communication systems and imaging. The thematic focus of the research in optofluidics is the interaction of fluids and light at the microscopic level. Overviews of this rapidly developing field are provided in [1,2].

One of the main topics in optofluidics is the development of optical elements that can be integrated on a microfluidic platform. Besides being an important component in so-called Lab-on-a-Chip devices [3], the use of a fluid medium brings with it advantages which could make such elements preferable to their solid-state counterparts for certain situations. Using a fluid medium makes it possible to dynamically reconfigure the element to adapt it to new operating conditions, for instance changing the focal length of a lens or the characteristic wavelength of a laser. Another advantage is the facility of achieving high-quality surfaces at low production cost. An example of a microfluidic waveguide, first described in [4], is shown in Fig. 1. Two fluids, with different refractive indices n_1 and n_2, flow in a microchannel. Light passes through the inner *core* fluid, which has a higher refractive index than the outer *cladding* fluid, and will thus be totally reflected if the angle of incidence is greater than the critical angle, which is determined by the ratio of n_1 to n_2.

The idea behind the waveguide can be extended to create a simple switch [4] (Fig. 2). In this case, the path of the core fluid, and hence the path of the light,

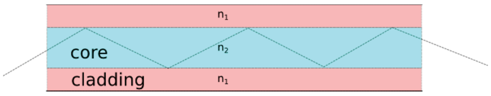


Fig. 1. Microfluidic waveguide based on the interface between two liquids.

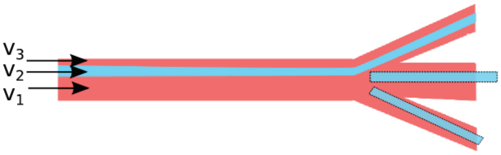


Fig. 2. Microfluidic switch controlled by the inflow speeds of the liquids.

is controlled by the relative flow rates of the cladding and the core fluids (v_1, v_3 and v_2, respectively).

Several other optical elements based on liquid-liquid interfaces have been investigated. Examples include lenses [5], gratings [6], and lasers [7].

2 Mathematical Model of Optical Elements

In the following, it is assumed that the influence of the light on the fluid flow is negligible so that the model for the optics can be separated from that of the flow. For characterization of the refraction properties of optical elements, a ray optics model can be used, since the typical length $L \approx 100\,\mu$m of the structures in the system is much larger than the wavelength of light $\lambda \approx 0.5\,\mu$m. For interference phenomena, a more complicated wave model would be required.

The incompressible Navier-Stokes equations provide a suitable description of the evolution of the flow which creates the optical elements, which is most often stationary and laminar. An important choice in the modeling is whether the fluids should be considered as immiscible or not. In most applications, the fluids do mix, which tends to smoothen out the boundary between them. This effect can be utilized to create Gradient Refractive Index lenses [8], but is not desirable in other cases. One can reduce the amount of mixing by raising the flow rates or using fluids with higher viscosities; and even eliminate it completely by choosing fluids which do not mix. In the latter case, one has to introduce an interface tension force $\boldsymbol{F}_\Gamma$ to model the interaction between the fluids. This force is defined only on the interface Γ and is proportional to the curvature κ and normal $\boldsymbol{n}$ of the interface. A material parameter σ depending on the two fluids determines the strength of the interaction, which is given by $\boldsymbol{F}_\Gamma = \sigma\kappa\boldsymbol{n}\delta_\Gamma$ [9].

The governing equations for the velocity $\boldsymbol{u}$ and pressure p are summarized in equation (1), where D represents the deformation tensor, ρ the density of the fluid and μ the viscosity. These equations have to be completed with suitable boundary and initial conditions depending on the concrete situation.

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\,\boldsymbol{u} &= \frac{1}{\rho}\left(-\nabla p + \nabla \cdot (2\mu D) + \boldsymbol{F}_\Gamma\right) \\ \nabla \cdot \boldsymbol{u} &= 0, \end{aligned} \tag{1}$$

The two material parameters ρ and μ are functions of both time and space, since they depend on how the interface Γ between the liquids evolves. A method of determining the interface is therefore required. Such an “interface model”, to be described in the next section, will deliver the quantities ρ, μ, κ and $\boldsymbol{n}$ to the flow model; which in turn will influence the interface through the evolution of the velocity field. The coupling of the two models is depicted schematically in Fig. 3.

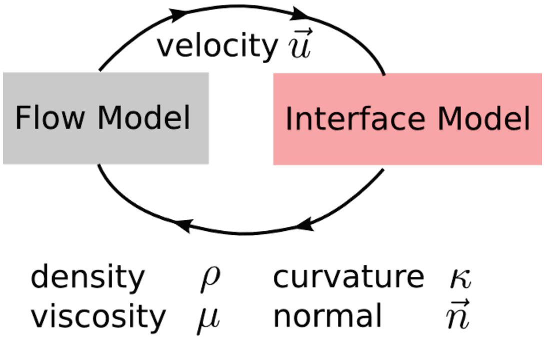


Fig. 3. Schematic view of the coupling between the flow and interface models.

3 Fluid Flow with Interface Tracking

The most common approaches for interface tracking can be divided into “tracking” methods and “capturing” methods, which differ in how the interface is represented. In the tracking methods, one uses an explicit representation in the form of a surface mesh, whose vertex coordinates $\boldsymbol{x}_i$ are evolved according the velocity field, e.g by the simple update $\boldsymbol{x}_i = \boldsymbol{x}_i + \boldsymbol{u}(\boldsymbol{x}_i)\Delta t$, with time-step Δt (see e.g. [10]).

The capturing methods, on the other hand, represent the interface implicitly using a function $\phi(\boldsymbol{x}, t)$ defined in the whole volume. ϕ can be chosen in different ways, but the capturing methods all have in common that the interface is represented as its zero-level contour: $\Gamma = \{\boldsymbol{x} | \, \phi(\boldsymbol{x}) = 0\}$. Like for the tracking methods, there is also an evolution equation for the interface in this case, in the form of an advection equation for ϕ:

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0.$$

The capturing methods have the advantage of being able to easily handle topological changes of the subdomains, whereas complicated algorithms (creating or destroying meshes) would be necessary for the tracking methods. Furthermore, tracking methods often require complicated mesh improvement or re-meshing procedures to maintain the quality of the mesh as time progresses. The price to pay for the capturing methods is solving an advection equation for the evolution of ϕ, which is in general difficult and computationally expensive. Furthermore, accurate capturing methods, such as the popular level set methods (see [9] for an overview), will require extra measures in order to guarantee mass conservation.

A widely used capturing method is the level set method described in [11], where ϕ is chosen to be a signed distance function, whose absolute value is defined by

$$|\phi(\boldsymbol{x}, t)| = \min_{\boldsymbol{y} \in \Gamma} |\boldsymbol{x} - \boldsymbol{y}|,$$

and whose sign at a point $\boldsymbol{x}$ is determined by what side of Γ the point lies on. ϕ has the attractive properties of being continuous and having a gradient of limited size $|\nabla\phi| = 1$. The latter property will not be conserved during the evolution of ϕ in time, unless a "re-initialization procedure" is performed, see [11]. A method to improve the mass conservation properties of the method has been proposed in [12].

An important decision when coupling any interface tracking method to a physical model is how to represent the quantities defined in the different domains separated by the interface, as well as the quantities that are only defined on the interface itself. In the example of the flow model, ρ and μ are discontinuous, and $\boldsymbol{F}_\Gamma$ must even be considered as a distribution, which might lead to numerical instabilities. It is therefore often prudent to regularize these quantities through smoothing even when doing so results in a less accurate method.

A study of this kind of regularization for modeling interface tension can be found in [13]. In that work, discontinuous quantities are smoothed out over a small width around the interface, and the interface tension is extended into the volume via the following extensions of the normal field and curvature:

$$\begin{aligned} \boldsymbol{n}(\boldsymbol{x}) &= \nabla\phi(\boldsymbol{x}), \\ \kappa(\boldsymbol{x}) &= -\nabla \cdot \boldsymbol{n}(\boldsymbol{x}). \end{aligned}$$

A suitable numerical method must be chosen to solve the advection equation for the level set function. Finite element methods based on discontinuous Galerkin or stabilized continuous Galerkin methods are appropriate choices for dealing with the discontinuities that can arise with this type of hyperbolic equation. In order to reduce computational costs, and still achieve an accurate representation of the interface, such methods should be combined with an adequate adaptive strategy.

References

1. Fainman, Y., Lee, L.P., Psaltis, D., Yang, C., eds.: Optofluidics. Fundamentals, Devices and Applications. McGraw-Hill (2010)
2. Hawkins, A.R., Schmidt, H., eds.: Handbook of Optofluidics. CRC Press (2010)
3. Chung, S.E., Park, W., Lee, S.A., Choi, S.E., Jang, J., Lee, S.H., Kwon, S.: Lab-on-a-chip. In Hawkins, A.R., Schmidt, H., eds.: Handbook of Optofluidics. CRC Press (2010)

4. Wolfe, D., Conroy, R., Garstecki, P., Mayers, B., Fischbach, M., Paul, K., Prentiss, M., Whitesides, G.: Dynamic control of liquid-core/liquid-cladding optical waveguides. Proceedings of the National Academy of Sciences of the United States of America **101** (2004) 12434–12438
5. Tang, S.K.Y., Stan, C.A., Whitesides, G.M.: Dynamically reconfigurable liquid-core liquid-cladding lens in a microfluidic channel. Lab on a Chip **8** (2008) 395–401
6. Hashimoto, M., Mayers, B., Garstecki, P., Whitesides, G.: Flowing lattices of bubbles as tunable, self-assembled diffraction gratings. Small **2** (2006) 1292–1298
7. Vezenov, D., Mayers, B., Conroy, R., Whitesides, G., Snee, P., Chan, Y., Nocera, D., Bawendi, M.: A low-threshold, high-efficiency microfluidic waveguide laser. Journal of the American Chemical Society **127** (2005) 8952–8953
8. Mao, X., Lin, S.C.S., Lapsley, M.I., Shi, J., Juluri, B.K., Huang, T.J.: Tunable liquid gradient refractive index (L-GRIN) lens with two degrees of freedom. Lab on a Chip **9** (2009) 2050–2058
9. Osher, S., Fedkiw, R.P.: Level set methods: An overview and some recent results. Journal of Computational Physics **169** (2001) 463 – 502
10. Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S., Jan, Y.J.: A front-tracking method for the computations of multiphase flow. Journal of Computational Physics **169** (2001) 708 – 759
11. Sussman, M., Smereka, P., Osher, S.: A level set approach for computing solutions to incompressible two-phase flow. Journal of Computational Physics **114** (1994) 146 – 159
12. Olsson, E., Kreiss, G.: A conservative level set method for two phase flow. Journal of Computational Physics **210** (2005) 225 – 246
13. Brackbill, J.U., Kothe, D.B., Zemach, C.: A continuum method for modeling surface tension. Journal of Computational Physics **100** (1992) 335 – 354

Generalized Polynomial Chaos in stochastic dynamical systems

Michael Schick

Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
michael.schick@kit.edu

Abstract. Generalized Polynomial Chaos (gPC) is known to fail for problems involving strong nonlinear dependencies on the stochastic input. Therefore, methods were developed to improve the convergence behavior of gPC, for example by decomposing the underlying probability space. This work focuses on a time–dependent approach (TD-gPC), where the basis functionals are allowed to evolve in time in a discrete way with respect to the probability distribution of the solution itself.

Keywords. Polyomial Chaos, Dynamical systems, Uncertainty quantification, Nonlinearity

1 Introduction

In 1938 Wiener [1] introduced the ”Homogeneous Chaos” based on Hermite–polynomials for Gaussian square integrable processes. It is a spectral projection method in the field of uncertainty quantification and was successfully applied in various fields of application, e.g. in combination with the Finite–Element–Method for problems in solid mechanics pioneered by Ghanem and Spanos in 1991 [2].

2 Spectral projection

2.1 Generalized Polynomial Chaos

As introduced by Xiu and Karniadakis in 2002 [3] a square–integrable stochastic process $X(t;\omega)$ can be decomposed into a summation of deterministic time dependent coefficients and stochastic basis functionals (Chaos Polynomials) denoted by

$$X(t;\omega) = \sum_{i=0}^{\infty} x_i(t)\psi_i(\zeta(\omega)), \tag{1}$$

whereas ζ denotes a random variable (possibly multi–dimensional) with a certain probability distribution. Furthermore, $\{\psi_i\}_{i=0}^{\infty}$ denotes a set of orthogonal polynomials, such that

$$\int \psi_i(\zeta)\psi_j(\zeta)\; dF_\zeta = C_i\delta_{ij}. \tag{2}$$

Here, F_ζ denotes the probability distribution function of ζ, $C_i > 0$ some constant and δ_{ij} the Kronecker–Delta. To achieve a feasible approximation of X, the summation in (1) is truncated at some integer $P > 0$ such that

$$X(t;\omega) \approx \sum_{i=0}^{P} x_i(t)\psi_i(\zeta(\omega)). \tag{3}$$

The (non–trivial) task left is to numerically compute the "modes" $x_i(t)$.

2.2 Time–dependent approach

The truncation of the sum in (3) is only capable of approximating nonlinear dependencies on ζ up to the order of P, which presents a significant drawback. Therefore, the key idea (based on [4]) is to improve the stochastic basis by letting the Chaos Polynomials depend on ζ and the solution itself. To this purpose, we fix the time $t = t^*$ and observe that a new random variable η can be defined via

$$\eta := \eta(\zeta(\omega)) = X(t^*;\omega) \approx \sum_{i=0}^{P} x_i(t^*)\psi_i(\zeta(\omega)). \tag{4}$$

Next we construct new orthogonal polynomials $\{\phi_i\}_{i=0}^{M}$ in terms of ζ and η and express the solution in the new basis via

$$X(t;\omega) = \sum_{i=0}^{M} x_i(t)\phi_i(\zeta(\omega), \eta(\omega)), \tag{5}$$

for $t \geq t^*$. Here, orthogonality means that the basis $\{\phi_i\}_{i=0}^{M}$ is orthogonal with respect to the cummulative probability distribution function of ζ and η. Note that the expansion order can change in this step and is therefore denoted by M.

Within the new basis it is expected that the solution as well as its dynamic evolution can be expressed in an optimal way. The procedure is then repeated at suitable times $t > t^*$.

2.3 Numerical results

We consider a linear oscillator [5], described by its equations of motion via

$$\frac{d}{dt}x_1(t;\omega) = x_2(t;\omega), \tag{6}$$

$$\frac{d}{dt}x_2(t;\omega) = -q(\zeta(\omega))x_1(t;\omega). \tag{7}$$

Here, ζ is a uniformly distributed random variable within the interval $(-1, 1)$ and $q(\zeta(\omega)) = 4\pi^2(1 + 0.2\zeta(\omega))$. For time discretization we employ a standard Runge Kutta solver of 4^{th} order with a time step size $\Delta t = 0.001$ and compare the results to the analytical solution of the problem. As can be seen in Fig. 1 TD-gPC is significantly increasing the accuracy in approximating the solution's stochastic moments in comparison to gPC.

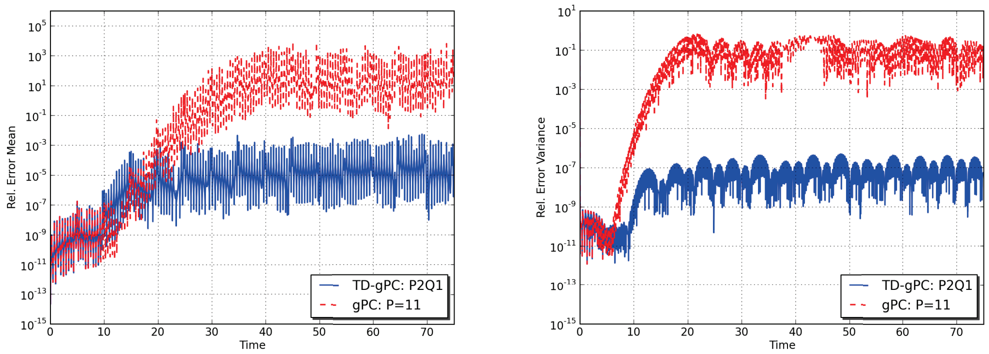


Fig. 1. Relative errors of mean (left) and variance (right) of x_1 for the time–dependent as well as standard gPC approach.

3 Conclusions

The time–dependent generalized Polynomial Chaos method provides a powerful tool to accurately capture the stochastic dynamics of a stochastic process. However, its price is an increase in the number of modes, which leads to high numerical costs. This increase is dependent on the size of the initial random vector ζ and the size of the underlying system of differential equations. Therefore, current research is focusing on modifications toward TD-gPC to overcome these drawbacks.

References

1. Wiener, N.: The homogeneous chaos. Am. J. Math. **60** (1938) 897–936
2. Ghanem, R.G., Spanos, P.D.: Stochastic Finite Elements: A Spectral Approach. Springer, New York (1991)
3. Xiu, D., Karniadakis, G.: The Wiener-Askey polynomial chaos for stochastic differential equations. SIAM Journal on Scientific Computing **24** (2002) 619–644
4. Gerritsma, M., van der Steen, J.B., Vos, P., Karniadakis, G.: Time-dependent generalized polynomial chaos. Journal of Computational Physics **229** (2010) 8333–8363
5. Le Maître, O.P., Mathelin, L.: Asynchronous time integration for polynomial chaos expansion of uncertain periodic dynamics. Discrete and Continuous Dynamical Systems **28** (2010) 199–226

Numerical Methods on Reconfigurable Hardware using High Level Programming Paradigms

Mareike Schmidtobreick

Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
mareike.schmidtobreick@student.kit.edu

Abstract. Recent developments, in particular in the field of reconfigurable hardware, open new possibilities related to numerical simulations which are in need of great accuracy and efficiency. In regard to applications for numerical purposes, two development paradigms, that convert high-level language code to a low-level hardware code which is then run on a *Field-Programmable Gate Array* (FPGA), are evaluated and compared. The handling of these converters and the performance of their generated code is evaluated. Eventually the numerical applicability is demonstrated by implementing three methods for solving systems of linear equations.

Keywords. DRC, FPGA, Impulse CoDeveloper, Reconfigurable Computing, ROCCC

1 Field Programmable Gate Arrays (FPGAs)

The architecture of a reconfigurable computing system usually contains one or more FPGAs. It is organized as either a cluster of FPGAs or as a hybrid system using a standard microprocessor CPU as the host system and an FPGA acting as an accelerator.

In contrast to conventional CPU processors, the configuration of an FPGA is not static but can be re-programmed. Further FPGAs consume a considerable lower amount of energy which results in significantly lower energy costs. An additional major advantage of FPGAs is that they can work highly parallel due to the massive spatial parallelism made possible by the configuration. However, some disadvantages apply when compared to a microprocessor, such as the low maximum clock frequency or the loss of logic due to configurability of the FPGA [1].

In general, each FPGA consists of the following three main configurable components: programmable logic, routing and I/O blocks.

Application areas for architectures involving FPGAs include bioinformatics (pattern-matching algorithms), image and signal processing (e.g. converting analog to digital signals (A/D input)) [2], and encryption and decryption algorithms.

In the field of supercomputing, FPGAs are not yet fully established as a standard used hardware. Algorithms such as Monte Carlo methods, Black-Scholes or financial analytics algorithms have already been successfully implemented and tested [2].
The process of mapping an algorithm to an FPGA differs from conventional programming and compilation. There are two ways of setting the FPGA configuration: One way is the classical low-level programming using a *Hardware Description Language* (HDL) in which all low-level choices have to be defined, e.g. clock behavior, design of data paths etc. [2] The other way is based on using *High-Level Languages* (HLL) which a compiler translates to HDL. They offer a higher level of abstraction than HDL.

After the HDL code is written, the algorithm has to undergo further steps, during which the experienced hardware programmer is able to influence the result and optimize performance by various factors. These three steps accomplished with the help of the Xilinx ISE tool include the synthesis, the “translate, map and 'place and route' step“ and finally the generation of the configuration file for the FPGA [3].

2 C-based FPGA programming with ROCCC

The *Riverside Optimizing Compiler for Configurable Computing* (ROCCC) 2.0 is an open source project. With its help, C-like code can be converted to *VHSIC hardware description language* (VHDL) code. But to be able to run the generated platform-independent VHDL code on an FPGA system, “glue-code” needs to be created that attaches to ROCCC's abstractions. The code design for ROCCC is a modular bottom-up design, distinguishing between module and system code. The modules can be thought of as building blocks only containing arithmetic expressions whereas system code builds these blocks together. Furthermore system code may include modules, may contain loops and therefore can iterate over arrays. There is a limitation on the use of data types: double precision is not supported. Furthermore division of floating point numbers has to be specially taken care of. In addition there are several constraints which need to be fulfilled. For example no loops are allowed except for-loops which themselves may not be doubly nested and no functions may be called other than further modules. For compilation ROCCC provides two scripts that have to be applied one after the other which results in VHDL code [4]. Unfortunately ROCCC does not provide the next compilation step for all platforms or equally detailed help.

3 C-based FPGA programming with *Impulse CoDeveloper*

Impulse CoDeveloper^{TM} (short *Impulse C*) is a development environment for hardware programming that includes a C cross-compiler, translation tools, an

IDE (Integrated Development Environment), various libraries and platform support packages. The programing model of *Impulse C* is based on the *CSP* (communicating sequential processes) programming model and is targeted at stream-oriented applications.

The C code is divided into two main parts: a software component (running on the CPU) and a hardware component (running on the FPGA). The software part includes the `main()` function, the hardware part two *Impulse C* specific functions `co_config` and `co_architecture` needed for initializations and declarations of *Impulse C* objects. Both parts contain so-called `co_functions` in which the intended calculations as well as the communication between software and hardware are implemented.

Impulse C specific objects include processes which run independently and are conceptually similar to programming with threads and communication objects such as streams, signals, semaphores and registers. Additionally a shared memory object between hardware and software processes is provided. There are some general constraints when writing C for hardware compilation (no recursion, limitations on conditional statements) and some constraints depending on the specific hardware platform (limitation of writing to memory from the CPU or the size of the on-board memory) being used.

The compilation process is relatively easy. *Impulse C* also offers a desktop simulation beforehand to help debugging and the "Application Monitor" to help visualize the different objects and their connections. After VHDL compilation moving over to the Xilinx ISE is simplified as a script is provided which allows the user to run the three steps necessary in Xilinx ISE through quite easily [5].

4 Performance Results

In order to find reasonable benchmarks, different tests were developed and analyzed. Tests are always run with different arithmetic operations using different sizes of input data. The time duration of the tests is measured using the function `gettimeofday()`.

The first three test-series have one software and one hardware process. In the first test-series P1 the software process reads n numbers from a file and sends them one by one to the hardware process. There an operation is carried out with a pair of numbers and the result is sent back to the software process. To omit the overhead of first reading from file, for the second test-series P2 numbers are generated randomly by the software process and directly sent to hardware. The performance results suggest that sending each number to hardware also outweighs the time for computation on the hardware side. Conclusively in P3 only a start and an iteration number are sent to hardware which then executes a given operation on the start number "iteration" times.

The result is that the division for floats is slower compared to addition or multiplication by the factor of about 5.8. Additionally, implementing floating point operations requires more logic of the FPGA than for fixed point numbers. Unfortunately when executing the operations in hardware the time the operations need is covered by the time the for-loop needs for execution.

In further tests the effect of changing the FPGAs frequency and duplicating the hardware processes and therefore partitioning the workload to several hardware processes were evaluated. Changing the frequency might result in wrong solutions and although duplicated hardware processes shorten the execution time, setting them up is laborious work.

A comparison of the FPGA-using application with a software-only version in respect to energy consumption leads to different results strongly depending on the structure and the parallelism of the applications. The best achieved results are given in Table 1.

Application n	**Total time of all executions**	**Power consumption of all executions**
Hybrid application	8m28s	31.63 Wh
Software-only application	11m22s	37.32 Wh

Table 1. Timing and energy consumption results of a hybrid application which is run 100 times and each time executes 10^8 iterations simultaneously in 10 hardware processes. Each hardware process contains a `for`-loop with one add-operation. These results are compared to a software-only version of it, containing the same number of calculations and `for`-loops.

5 Numerical Solutions

The considered problem is the Poisson equation with homogeneous Dirichlet boundary conditions. For the discretization we use the Finite Difference method which leads to a system of linear equations $A_h u_h = b_h$ [6]. Such a system of linear equations can be solved with the Conjugate Gradient (CG) method and to improve performance we consider applying a preconditioner. Possible preconditioners include one of the following splitting methods: the Jacobi, the Gauss-Seidel or the SOR method. To evaluate numerical applicability of HLL to VHDL converters the Jacobi, Gauss-Seidel and SOR method are implemented using *Impulse C* and its associated tools for the FPGA. The main focus is laid on of utilizing them as preconditioners to the CG method [6,7].

The first step for each of these algorithms is to decide what part of the algorithm is to be implemented on the software and which part on the hardware side (in the associated two processes). As these algorithms are included into the CG method as a preconditioner, there is no need to calculate the residual anymore, as the CG method already does this. As a consequence only the calculation of the new correction is outsourced to hardware. As the residual is needed for calculation in both hardware and software parts, it is stored in the shared memory accessible for all processes, while signals guarantee the exclusive access rights.

For the Gauss-Seidel method and the SOR method to improve performance the red-black method is used which leads to a re-ordering of the mesh points and permits parallelization (see [8] for more details). Timing results are given in Table 2.

	6	50	100	1,000	*sdim*
Jacobi method	0.001515	0.181461	0.756125	76.964983	time in [s]
Gauss-Seidel method	0.001520	0.184684	0.739551	79.768941	time in [s]

Table 2. Timing results for the Gauss-Seidel and the SOR method with a fixed number of iterations $k_{max} = 20$ and for a different number of points on the discretization mesh (with *sdim* giving the number of points on one edge of the discretized square domain).

6 Conclusion

ROCCC is a compiler working with C code that is very close to standard ANSI-C. However for a newcomer the help of an expert is required in order to complete the last step of mapping the VHDL code to the actual FPGA.

The structure and programming model of Impulse C and its Impulse C elements take some time to become acquainted with. For experienced users, the manual instructions provide an opportunity to influence the generation procedure if necessary and to optimize the application on their own. But to be able to truly optimize the application, the user needs to understand the underlying technology and the user has to know how the different operations are implemented on the FPGA.

The performance of the FPGA highly depends on the properties of the application and especially the implementation concept. If an application can be easily parallelized and most computations of data do not depend on other data, then the application executed on a hybrid system most likely outperforms the CPU. As a general conclusion, the usage of Impulse C is recommended for new users that want to run user-written code on an FPGA, although this for new users helpful abstraction level may lead to a lack of performance. ROCCC should be used as a VHDL generation tool by more experienced users. It is still evolving and has a promising potential.

References

1. Mitrionics: Low Power Hybrid Computing for Efficient Software Acceleration 1.0.0-001 (2008)
2. Gokhale, Maya; Graham, P.S.: Reconfigurable computing : accelerating computation with field-programmable gate arrays. Springer, Berlin [u.a.] (2005)

3. Wain, R., Bush, I., Guest, M., Deegan, M., Kozin, I., Kitchen, C.: An overview of FPGAs and FPGA programming; Initial experiences at Daresbury. Technical report, Computational Science and Engineering Department, CCLRC Daresbury Laboratory, Daresbury, Warrington, Cheshire, WA4 4AD, UK (2006) Version 2.0.
4. Anon.: ROCCC 2.0 Developer's Manual - Revision 0.4.2 (2010)
5. Anon.: CoDeveloper User Guide (2009) Version 3.60.
6. Knabner, Peter ; Angermann, L.: Numerik partieller Differentialgleichungen : eine anwendungsorientierte Einführung. Springer, Berlin (2000)
7. Ueberhuber, C.W.: Numerical Computation 2. Volume 2:. Springer, Berlin (1997)
8. Schwandt, H.: Parallele Numerik : eine Einführung. 1. aufl. edn. Lehrbuch : Mathematik. Teubner, Stuttgart (2003)

Parallel Criticality Calculations with Parafish

Chandramowli Subramanian[1], Serge Van Criekingen[2]

[1] Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
chandramowli.subramanian@kit.edu
[2] Karlsruhe Institute of Technology,
Institute for Neutron Physics and Reactor Technology
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany
serge.criekingen@kit.edu

Abstract. The Davidson method is implemented within the neutron transport core solver *Parafish* to solve k-eigenvalue criticality transport problems. The *Parafish* solver is based on domain-decomposition, uses spherical harmonics (P_N method) for angular discretization, and non-conforming finite elements for spatial discretization. The Davidson method is compared to the traditional power method in that context. Encouraging numerical results are obtained.

Keywords. criticality, neutron transport, eigenvalue problems, parallel computing

1 Introduction

Solving the Boltzmann Transport Equation (BTE) is a key issue in neutronics computations. With the advent of high performance computing, the development of parallel algorithms to solve the BTE has become a must to take advantage of large processor clusters. Moreover, even basic desktop computers are now equipped with several processors whose collaborative use can lead to significant speedups. Therefore, the importance of purely sequential solvers is meant to decrease rather drastically within the next decade. In this view, a parallel solver named *Parafish* [14] is currently being developed as a collaboration between the *Institute for Neutron Physics and Reactor Technology* and *Engineering Mathematics and Computing Lab*. The development of *Parafish* has been done in parallel starting from the very beginning – it is not the parallelized version of a pre-existing code.

In view of the increasing computing capabilities made available with high performance computing, we anticipate that more advanced approximation schemes than diffusion will be used in the future to treat the BTE. The spherical harmonics (or P_N) method goes beyond the diffusion approximation, and has been implemented in *Parafish*.

2 The Criticality Problem

The behavior of neutrons within a nuclear reactor core, that is the well-known fission chain reaction, is modeled by the BTE. Evaluating the stability of a reactor is mathematically done by means of an eigenvalue calculation based on this equation. The most widely used form of this kind of eigenvalue calculation is known as the k-eigenvalue criticality calculation. It seeks for the largest eigenvalue k and the angular flux $\psi(\mathbf{r}, \mathbf{\Omega}, E)$ of

$$\begin{aligned}\Big[\mathbf{\Omega}\cdot\nabla + \sigma(\mathbf{r},E)\Big]\psi(\mathbf{r},\mathbf{\Omega},E) = \\ \int_0^\infty \int_{\mathcal{S}} \sigma_s(\mathbf{r},E'\rightarrow E,\mathbf{\Omega}'\cdot\mathbf{\Omega})\,\psi(\mathbf{r},\mathbf{\Omega}',E')\,d\mathbf{\Omega}'\,dE' \\ +\frac{\chi(E)}{k}\int_0^\infty\int_{\mathcal{S}}\nu(E')\,\sigma_f(\mathbf{r},E')\,\psi(\mathbf{r},\mathbf{\Omega}',E')\,d\mathbf{\Omega}'\,dE',\end{aligned} \tag{1}$$

see e.g. [7]. Here, $\sigma(\mathbf{r}, E)$ is the macroscopic total cross-section (assumed strictly positive), $\sigma_s(\mathbf{r}, E' \rightarrow E, \mathbf{\Omega}' \cdot \mathbf{\Omega})$ the scattering cross-section, and $\sigma_f(\mathbf{r}, E')$ the fission cross-section. Moreover, $\chi(E)$ is the spectrum of emitted fission neutrons and $\nu(E')$ the number of neutrons emitted per fission.

Stability is ensured when the largest eigenvalue k is equal to one. Then, the fission chain reaction is self-sustained. A k-eigenvalue larger than one indicates an exponential multiplication of the neutron population within the reactor. Similarly, a k-eigenvalue smaller than one indicates that less neutrons are produced than absorbed, so that the chain reaction dies out.

The power method, sometimes accelerated with Chebyshev acceleration, has been up to now the method of choice for solving the k-eigenvalue criticality problem in reactor applications [7]. Several research works have been conducted in the past to find a better alternative to this rather basic method. In this view, several ways have been investigated: the implicit restarted Arnoldi method [1,16,17], the Jacobi-Davidson method [5,15], and, more recently, the Jacobian-free Newton-Krylov method [3,4,6]. These techniques have been up to now applied to the diffusion equation or to the discrete ordinate (S_N) transport approximation only.

For the criticality calculations we investigate the Davidson method [2,10] which is in fact an earlier development of the Jacobi-Davidson method [11]. We believe that such an investigation is valuable since the Jacobi-Davidson is not proved to be better than Davidson in all cases [8].

3 Numerical Results

We consider the 2-D MOX fuel assembly benchmark issued by the NEA [9]. This benchmark uses C5 MOX fuel and 7 energy groups, hence its "C5G7" nickname.

The numerical results with the spherical harmonic discretization P_3 obtained by employing the *Parafish* solver are given in Table 1 (see [12,13]). The computing times as well as the number of iterations are drastically reduced by using the Davidson method. This is due to the fact that the Davidson method keeps track of more information along the iterative procedure, the extra storage cost being outweighted by the reduction of the number of iterations.

	Davidson Method			Power Method		
# iterations	19			37		
# cores	Time (s)	Speedup	Efficiency	Time (s)	Speedup	Efficiency
32	161	1.	1.	528	1.	1.
64	96	1.68	0.84	325	1.62	0.81
256	30	5.41	0.68	100	5.28	0.66
400	20	8.00	0.64	72	7.38	0.59

Table 1. Results for the C5G7 benchmark with the P_3 approximation (14,962,584 unknowns), see [12,13]. The inner iterations refer to GMRES iterations in both cases. Speedups and efficiencies are computed with respect to the 32 core case.

References

1. Cao, Y., and Lee, J. C. A Krylov subspace method to calculate k- and α-eigenvalues. In *Proceedings of the PHYSOR 2008 International Conference on the Physics of Reactors "Nuclear Power: A Sustainable Resource"* (Interlaken, Switzerland, September 14-19 2008).
2. Crouzeix, M., Philippe, B., and Sadkane, M. The Davidson method. Tech. rep., IRISA - Institut de Recherche en Informatique et Systèmes Aléatoires, November 1990. Publication interne No 558.
3. Gill, D. F., and Azmy, Y. Y. Jacobian-free Newton-Krylov as an alternative to power iterations for the k-eigenvalue transport problem. *ANS Transactions 92* (2005), 728–730.
4. Gill, D. F., and Azmy, Y. Y. A Jacobian-free Newton-Krylov iterative scheme for criticality calculations based on the neutron diffusion equation. In *Proceedings of the M&C 2009 International Conference on Mathematics, Computational Methods & Reactor Physics* (Saratoga Springs, New York, USA, May 3-7 2009).
5. Havet, M. *Solution of Algebraic Problems arising in Nuclear Reactor Simulations using Jacobi-Davidson and Multigrid Methods.* PhD thesis, Université Libre de Bruxelles, Brussels, Belgium, 2008.
6. Knoll, D. A., Park, H., and Smith, K. Application of the Jacobian-Free Newton-Krylov method in computational reactor physics. In *Proceedings of the M&C 2009 International Conference on Mathematics, Computational Methods & Reactor Physics* (Saratoga Springs, New York, USA, May 3-7 2009).
7. Lewis, E. E., and Miller, Jr., W. *Computational Methods of Neutron Transport.* John Wiley & Sons, 1984.

8. Notay, Y. Is Jacobi-Davidson faster than Davidson? *SIAM J. Matrix Anal. Appl. 26* (2005), 522–543.
9. OECD/NEA. Benchmark on deterministic transport calculations without spatial homogenisation. A 2-D/3-D MOX fuel assembly benchmark. vol. NEA/NSC/DOC(2003)16 of *Nuclear Science*, OECD, Nuclear Energy Agency.
10. Sadkane, M. Block Arnoldi and Davidson methods for unsymmetric large eigenvalue problems. *Numer. Math. 64* (1993), 195–211.
11. Sleijpen, G., and van der Vorst, H. A Jacobi-Davidson iteration for linear eigenvalue problems. *SIAM Review 42*, 2 (2000), 267–293.
12. Subramanian, C. High performance computing for stability problems – Applications to hydrodynamic stability and neutron transport criticality. Dissertationsschrift, Karlsruhe Institute of Technology, 2011.
13. Subramanian, C., Van Criekingen, S., Heuveline, V., Nataf, F., and Havé, P. The Davidson method as an alternative to power iterations for criticality calculations. Submitted for publication.
14. Van Criekingen, S., Nataf, F., and Havé, P. Parafish: a parallel $FE - P_N$ neutron transport solver based on domain decomposition. *Annals of nuclear energy 38*, 1 (2011), 145–150.
15. Verdú, G., Ginestar, D., Miró, R., and Vidal, V. Using the Jacobi-Davidson method to obtain the dominant lambda modes of a nuclear power reactor. *Annals of nuclear energy 32* (2005), 1274–1296.
16. Verdú, G., Miró, R., Ginestar, D., and Vidal, V. The implicit restarted Arnoldi method, an efficient alternative to solve the neutron diffusion equation. *Annals of nuclear energy 26* (1999), 579–593.
17. Warsa, J. S., Wareing, T. A., Morel, J. E., McGhee, J. M., and Lehoucq, R. B. Krylov subspace iterations for deterministic k-eigenvalue calculations. *Nuclear Science and Engineering 147*, 1 (2004), 26–42.

A highly efficient Scalable Solver for the Global Ocean/Sea-Ice Model MPIOM

Florian Wilhelm[1], Panagiotis Adamidis[2] and Vincent Heuveline[1]

[1] Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
Florian.Wilhelm@kit.edu, Vincent.Heuveline@kit.edu
[2] Scientific Computing, Deutsches Klimarechenzentrum GmbH
20146, Hamburg, Bundesstraße 45a, Germany
Adamidis@dkrz.de

Abstract. We show how to efficiently solve the barotropic subsystem of an ocean model based on the primitive ocean equations. This is accomplished by replacing the traditionally often used Successive-Over-Relaxation method by the Conjugate Gradient method with an Incomplete Cholesky preconditioner with fill-in p.

Keywords. HPC, SOR, CG method, ICC(p), MPIOM, climate simulation, mathematical modeling, ocean model, barotropic subsystem

1 Model Description

The Max-Planck-Institute Ocean Model MPIOM is an Ocean General Circulation Model (OGCM) based on the ocean primitive equations on a curvilinear C-grid with z-coordinates and free surface [1]. It includes a sea-ice model as well as the HAMOCC biogeochemical submodel and was released in 1997 as successor of the Hamburg Ocean Primitive Equation Model (HOPE) [2]. Together with the atmosphere model ECHAM it was used in the Intergovernmental Panel for Climate Change (IPCC) Assessment Report 4 and is currently used in preparation of the IPCC Assessment Report 5.

2 Horizontal Conservation of Momentum

The horizontal conservation of momentum equation for a Boussinesq fluid on a rotating sphere with orthogonal coordinates is given as

$$\frac{d}{dt}\mathbf{u} + f(\mathbf{z} \times \mathbf{u}) = -\frac{1}{\rho_w}\nabla_H(p + \rho_w g\eta) + \mathbf{F}_H + \mathbf{F}_V,$$

where $\mathbf{u} = (u, v)$ are the horizontal components of velocity $\mathbf{v}$, t is the time, f the Coriolis parameter, $\mathbf{z}$ the vertical unit vector, ρ_w the reference density of sea-water, ∇_H the horizontal gradient operator, p the internal pressure, g

the effective gravitational acceleration and η the sea-surface elevation. The total derivative is $\frac{d}{dt} = \partial_t + \mathbf{u} \cdot \nabla_H + w \cdot \partial_z$, where w is the vertical component of $\mathbf{v}$. The terms $\mathbf{F}_H$ and $\mathbf{F}_V$ describe the horizontal and vertical eddy viscosity. This equation is solved for $\mathbf{u}$, p and η. More details regarding the governing equations in MPIOM can be found in [1].

Decomposing $\mathbf{u}$ following [3] into its barotropic $\bar{\mathbf{u}}$ and baroclinic $\tilde{\mathbf{u}}$ components

$$\bar{\mathbf{u}} = \int_{-H}^{0} \mathbf{u}\, dz, \qquad \tilde{\mathbf{u}} = \mathbf{u} - \frac{\bar{\mathbf{u}}}{H}$$

results in a *baroclinic subsystem*

$$\partial_t \tilde{\mathbf{u}} + A(\mathbf{u},\mathbf{u}) - \bar{A}(\mathbf{u},\mathbf{u}) + f(\mathbf{z} \times \tilde{\mathbf{u}}) = \frac{1}{H} \int_{-H}^{0} \nabla_H \hat{p}\, dz - \nabla_H \hat{p} + \tilde{\mathbf{F}}_H + \tilde{\mathbf{F}}_V,$$

and a *barotropic subsystem*

$$\partial_t \bar{\mathbf{u}} + \bar{A}(\mathbf{u},\mathbf{u}) + f(\mathbf{z} \times \bar{\mathbf{u}}) = -gH\nabla_H \eta - \int_{-H}^{0} \nabla_H \hat{p}\, dz + \bar{\mathbf{F}}_H + \bar{\mathbf{F}}_V,$$

with $\hat{p} = p/\rho_w$, $A(\mathbf{u},\mathbf{u}) = (\mathbf{u} \cdot \nabla)\mathbf{u} + w\partial_z \mathbf{u}$ and analogously defined barotropic $\bar{A}(\mathbf{u},\mathbf{u})$, $\bar{\mathbf{F}}_H$, $\bar{\mathbf{F}}_V$ as well as baroclinic $\tilde{\mathbf{F}}_H$, $\tilde{\mathbf{F}}_V$. Partially updating $\bar{\mathbf{u}}$ with respect to $A(\mathbf{u},\mathbf{u})$ and $\bar{\mathbf{F}}_H$, $\bar{\mathbf{F}}_V$ in the barotropic subsystem and using $\hat{p}$ from the solved baroclinic subsystem together with the barotropic continuity equation

$$\partial_t \eta + \partial_x \bar{u} + \partial_y \bar{v} = 0,$$

results after discretization in a linear equation system for η, that is symmetric positive definite with non-positive off-diagonal entries. This is a so called Stieltjes matrix with shape

$$A = \begin{pmatrix} D_1 & S_1 & & & \\ S_1 & D_2 & S_2 & & \\ & S_2 & \ddots & \ddots & \\ & & \ddots & \ddots & S_n \\ & & & S_n & D_n \end{pmatrix},$$

where S_i are diagonal matrices and D_i tridiagonal matrices. The matrix A can also be expressed as a stencil as shown in Figure 1.

3 Solving the Barotropic System

The system was traditionally solved with the Successive-Over-Relaxation (SOR) method

$$(D + \omega L)x_{k+1} = ((1-\omega)D - \omega L^T)x_k + \omega b,$$

where D is the diagonal of A, L its strict lower part, $U = L^T$ its strict upper part and ω a relaxation parameter.

The SOR method was replaced by CG Method with ICC(p) during the ScalES project in order to achieve better performance and scalability.

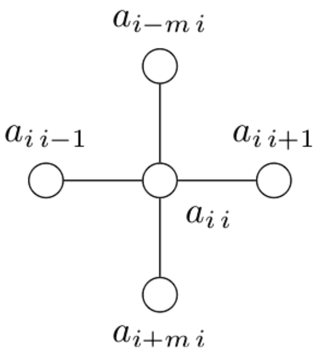


Fig. 1. Five-point stencil of the barotropic subsystem

4 The Incomplete Cholesky Preconditioner with Fill-in *p*

To efficiently implement an Incomplete Cholesky Preconditioner (ICC(p)), that is $LL^T \approx A$, the fill-in pattern of the stencil in Figure 1 was analyzed. One can show that the fill-in pattern obeys

$$l_{ij}^p = 0 \text{ if } l_{ij}^p \notin \text{diag}_k(L) \text{ with } k = 0, \ldots, \max(1, p-1), m-p, \ldots, m.$$

It was therefore decided to implement the ICC(p) preconditioner in the spirit of the DIAG format [4]. The lower tridiagonal matrix l can also be expressed as a stencil:

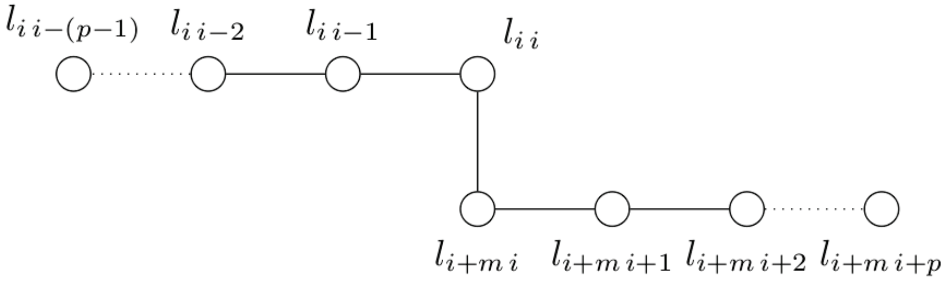


Fig. 2. Stencil of the ICC(p) factorization.

Benchmarks with the Hardware Performance Monitor (HPM) showed a cache hit rate of 99.945% with our implementation based on this analysis.

5 Benchmark Results

As benchmark we used the TP6ML20 model which is based on a tripolar grid with resolution of 0.1°. That makes 3602×2394 grid points with about 8.6 million unknowns. A relative residual of 10^{-11} and 0 as initial vector were chosen.

All tests were performed on the Blizzard (DKRZ) super computer based on IBM p575 POWER6 system with 264 nodes with 16 dual core CPUs per node. This makes a total of 8448 cores with a peak performance of 158 Teraflop/s coupled with an InfiniBand Fat CLOS Tree interconnect with 7.6 Terabyte/s.

The achieved results are overly promising as can be seen in Figure 3. The overall scalability of the Conjugate Gradient Method with ICC(4) is up to 36% better than the traditional SOR solver on the largest setup.

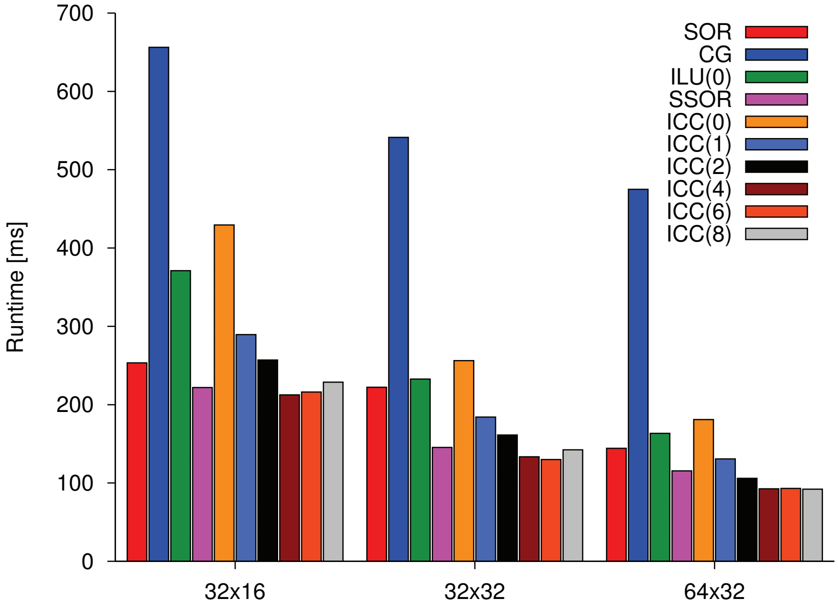


Fig. 3. Runtime of CG-Solver with preconditioners. Rel. residual 1e-11

References

1. Marsland, S., Haak, H., Jungclaus, J., Latif, M., Röske, F.: The Max-Planck-Institute Global Ocean/Sea Ice Model with Orthogonal Curvilinear Coordinates. Ocean Modelling **5** (2003) 91–127
2. Wolff, J., Maier-Reimer, E., Legutke, S.: The Hamburg Ocean Primitive Equation Model HOPE. Technical report, DKRZ Hamburg (1997)
3. Simonnet, E., Medjo, T.T., Temam, R.: Barotropic-baroclinic formulation of the primitive equations of the ocean. Applicable Analysis **82** (2003) 439 – 456
4. Saad, Y.: Iterative methods for sparse linear systems. 2. ed. edn. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, PA (2003)

Model Reduction for Instationary Flow Problems with the Proper Orthogonal Decomposition Method

Martin Wlotzka

Karlsruhe Institute of Technology, Engineering Mathematics and Computing Lab
Fritz-Erler-Str. 23, 76133 Karlsruhe, Germany
martin.wlotzka@kit.edu

Abstract. With the proper orthogonal decomposition method, reduced models of complex processes can be established. A reduced model is characterized by much lower number of degrees of freedom compared to the original model established e.g. by a finite element discretisation or observation data. The reduction is achieved by the use of problem-specific basis functions. The method is applied to an instationary channel flow problem. Quality and optimization of the reduced model are investigated.

Keywords. proper orthogonal decomposition, model reduction, model adaptivity, optimal snapshot location

1 Proper Orthogonal Decomposition (POD)

In the snapshot-form of the POD method one assumes the availability of functions in the spatial variable, which represent the solution of the underlying system at certain instants of time. These functions are called snapshots. With the snapshots, a so-called POD basis of a function space of low dimension can be computed. The POD basis $\{\psi_1, ..., \psi_l\}$ is the solution of the minimization problem

$$\begin{aligned} \min \sum_{j=1}^{m} \left\| y_j - \sum_{i=1}^{l} \langle y_j, \psi_i \rangle \psi_i \right\|^2 \\ \text{s.t.} \quad \langle \psi_i, \psi_k \rangle = \delta_{ik} \quad (1 \leq i, k \leq l) \ , \end{aligned}$$

where the snapshots $y_1, ..., y_m$ are elements of some Hilbert space. Here, the snapshots are assumed to be linearly independent and $l \leq m$ holds [1].

2 Model Reduction

Model reduction is achieved by using the information contained in the ensemble of snapshots. A two-dimensional instationary channel flow around a cylinder serves as an application for model reduction. The flow is governed by the instationary Navier-Stokes equations. Out of the initial flow of the incompressible Newtonian fluid, the Kármán vortex street evolves, and the solution turns into a time-periodic state [2]. The snapshots are taken from a finite element approximation of the solution. For details on the POD method, the flow problem, the finite element discretisation and the model reduction see [3]. The methods are implemented in the HiFlow^3 software package [4].

The POD basis functions are used as ansatz and test functions for a Galerkin approximation. The flow problem is solved in the reduced space. The quality of a reduced model is measured in terms of the error

$$\|u_h - u_p\| \ ,$$

where u_h is the finite element approximation and u_p is the solution of the reduced model. Having an ensemble of snapshots, it turnes out that the choice of the snapshots taken into account for computing a POD basis as well as the rank of the POD basis are of essential relevance for the quality of the reduced model. Taking the snapshots only out of the interval of the first period of the flow, no accurate reduced model can be obtained, since the snapshots contain no information on the initial flow phase. Chosing snapshots from the whole time interval in contrast, it is possible to obtain accurate reduced models, if the rank of the POD basis is chosen high enough. A reduced model with 50 degrees of freedom can be stablished, which yields an approximate solution with a relative error of less than one percent. The number of degrees of freedom is reduced by three orders of magnitude compared to the finite element discretisation. The four exemplary POD basis functions in Fig. 1 represent some major features of the flow. The vortex structure is contained in the two basis functions on the left, and the two basis functions on the right contain structures corresponding to the initial flow.

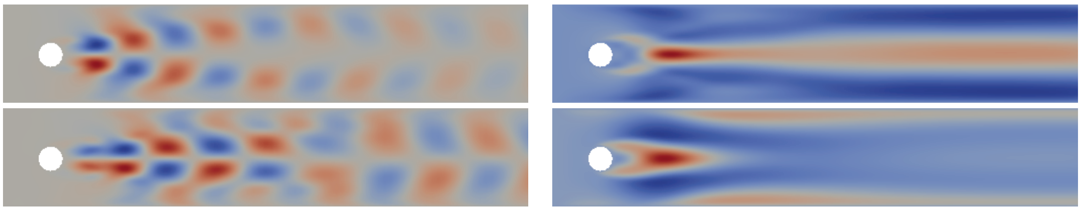


Fig. 1. Exemplary POD basis funtions (x-component of the velocity).

3 Adaptivity of the Reduced Model - Optimal Snapshot Location

The cost for computing a POD solution is essentially determined by the number of snapshots and the rank of the POD basis. When the number of snapshots and the rank of the POD basis are fixed, the quality of the reduced model is determined by the location of the snapshots. Adaptive methods to determine the optimal choice of the snapshot location can be developed by means of an optimization problem. Optimality conditions of first and second order can be derived and an algorithm for the computation of the optimal snapshot location is given in [5] and further developed for the nonlinear channel flow problem in [3]. The results show, that accurate reduced models with few snapshots and a low rank of the POD basis can be obtained. For a reduced model with only 30 snapshots and 6 POD basis functions an optimal snapshot location yields an approximate solution with a relative error of less than three percent.

References

1. Volkwein, S.: Optimal Control of a Phase Field Model Using Proper Orthogonal Decomposition. ZAMM **Vol. 81, No. 2** (2001) 83–97
2. Turek, S., Schäfer, M.: Benchmark computations of laminar flow around cylinder. Hirschel, E.H.: Notes on Numerical Fluid Mechanics **52** (1996) 547–566 co. F. Durst, E. Krause, R. Rannacher.
3. Wlotzka, M.: Modellreduktion instationärer Strömungsvorgänge mit der Proper Orthogonal Decomposition Methode. Diplomarbeit, Karlsruhe Institute of Technology, Department of Mathematics (2010)
4. Anzt, H., Augustin, W., Baumann, M., Bockelmann, H., Gengenbach, T., Hahn, T., Heuveline, V., Ketelaer, E., Lukarski, D., Otzen, A., Ritterbusch, S., Rocker, B., Ronnas, S., Schick, M., Subramanian, C., Weiss, J.P., Wilhelm, F.: HiFlow3 – A Flexible and Hardware-Aware Parallel Finite Element Package. EMCL Preprint Series 2010-06 (ISSN 2191-0693)
5. Kunisch, K., Volkwein, S.: Optimal snapshot location for computing POD basis functions. ESAIM: M2AN **44** (2010) 509–529

Preprint Series of the Engineering Mathematics and Computing Lab

recent issues

No. 2011-11 Thomas Gengenbach, Vincent Heuveline, Mathias J. Krause: Numerical Simulation of the Human Lung: A Two–scale Approach

No. 2011-10 Vincent Heuveline, Dimitar Lukarski, Fabian Oboril, Mehdi B. Tahoori, Jan-Philipp Weiss: Numerical Defect Correction as an Algorithm-Based Fault Tolerance Technique for Iterative Solvers

No. 2011-09 Vincent Heuveline, Dimitar Lukarski, Nico Trost, Jan-Philipp Weiss: Parallel Smoothers for Matrix-based Multigrid Methods on Unstructured Meshes Using Multicore CPUs and GPUs

No. 2011-08 Vincent Heuveline, Dimitar Lukarski, Jan-Philipp Weiss: Enhanced Parallel ILU(p)-based Preconditioners for Multi-core CPUs and GPUs – The Power(q)-pattern Method

No. 2011-07 Thomas Gengenbach, Vincent Heuveline, Rolf Mayer, Mathias J. Krause, Simon Zimny: A Preprocessing Approach for Innovative Patient-specific Intranasal Flow Simulations

No. 2011-06 Hartwig Anzt, Maribel Castillo, Juan C. Fernández, Vincent Heuveline, Francisco D. Igual, Rafael Mayo, Enrique S. Quintana-Ortí: Optimization of Power Consumption in the Iterative Solution of Sparse Linear Systems on Graphics Processors

No. 2011-05 Hartwig Anzt, Maribel Castillo, José I. Aliaga, Juan C. Fernández, Vincent Heuveline, Rafael Mayo, Enrique S. Quintana-Ortí: Analysis and Optimization of Power Consumption in the Iterative Solution of Sparse Linear Systems on Multi-core and Many-core Platforms

No. 2011-04 Vincent Heuveline, Michael Schick: A local time–dependent Generalized Polynomial Chaos method for Stochastic Dynamical Systems

No. 2011-03 Vincent Heuveline, Michael Schick: Towards a hybrid numerical method using Generalized Polynomial Chaos for Stochastic Differential Equations

No. 2011-02 Panagiotis Adamidis, Vincent Heuveline, Florian Wilhelm: A High-Efficient Scalable Solver for the Global Ocean/Sea-Ice Model MPIOM

No. 2011-01 Hartwig Anzt, Maribel Castillo, Juan C. Fernández, Vincent Heuveline, Rafael Mayo, Enrique S. Quintana-Ortí, Björn Rocker: Power Consumption of Mixed Precision in the Iterative Solution of Sparse Linear Systems

No. 2010-07 Werner Augustin, Vincent Heuveline, Jan-Philipp Weiss: Convey HC-1 Hybrid Core Computer – The Potential of FPGAs in Numerical Simulation

No. 2010-06 Hartwig Anzt, Werner Augustin, Martin Baumann, Hendryk Bockelmann, Thomas Gengenbach, Tobias Hahn, Vincent Heuveline, Eva Ketelaer, Dimitar Lukarski, Andrea Otzen, Sebastian Ritterbusch, Björn Rocker, Staffan Ronnås, Michael Schick, Chandramowli Subramanian, Jan-Philipp Weiss, Florian Wilhelm: HiFlow^3 – A Flexible and Hardware-Aware Parallel Finite Element Package

The responsibility for the contents of the working papers rests with the authors, not the Institute. Since working papers are of a preliminary nature, it may be useful to contact the authors of a particular working paper about results or caveats before referring to, or quoting, a paper. Any comments on working papers should be sent directly to the authors.