

PDESofT 2014

HEIDELBERG, JULY 14–16, 2014
BOOK OF ABSTRACTS

Scientific Committee

Peter Bastian
Andreas Dedner
Christian Engwer

Guido Kanschat
Hans Petter Langtangen
Randall J. LeVeque

Version: July 4, 2014

Contents

1	Keynote lectures	3
1.1	Surviving and thriving as an open scientific software project	4
1.2	Large-Eddy Simulation with OpenFOAM: Practice, Applications and Future Challenges	5
1.3	Modeling waves with PyClaw: design, development, and applications	6
1.4	Firedrake and dolfin-adjoint: Defeating the complexity barrier in simulation generation by composing abstractions	7
2	Contributed lectures	8
2.1	HiFlow ³ – A Hardware-Aware Parallel Finite Element Package	9
2.2	PyOP2: a performance-portable framework for parallel computations on unstructured meshes	10
2.3	Recent Development in NGSolve	11
2.4	Experiences using 2decomp&fft to solve Partial Differential Equations using Fourier Spectral Methods	12
2.5	Introduction to hpGEM: Software library for discontinuous Galerkin methods	13
2.6	The BEM++ boundary element library	14
2.7	DOP-ELIB: The differential equations and optimization environment	15
2.8	Extruded Meshes for High Aspect Ratio Simulations in Firedrake and PyOP2	16
2.9	Automated Build System for PDE Toolboxes	17
2.10	Entropy-based viscous regularization for the multi-dimensional Euler equations in low-Mach regimes	18
2.11	pyMOR – Model Order Reduction with Python	19
2.12	Interactive Simulations using Parallel and Event-Driven Localized Reduced Basis	20
2.13	Abstraction of Multiscale Methods: Theory and Implementation	21
2.14	An Adaptive Framework for a Heterogeneous Multiscale Method	22
2.15	Solving PDEs in High Space Dimension using DUNE	23
2.16	LifeV: parallel framework and applications to blood-flow simulations	24
2.17	Domain decomposition based fluid-structure interaction algorithm based on Steklov-Poincaré operators	25
2.18	HiFlow ³ -based Numerical Simulations for Cognition-Guided Surgery	26

2.19	IsoGlib: an Isogeometric Analysis library for the solution of high-order Partial Differential Equations on surfaces	27
2.20	TEMPO - a one step alternative to SUNDIALS	28
2.21	PetIGA: High-Performance Isogeometric Analysis	29
2.22	Locally adaptive space-time grid in electromagnetic plasma simulations	30
2.23	A computational approach for problems involving moving interfaces	31
2.24	A New Interface-based Approach to Heterogeneous Grid Coupling in DUNE	32
2.25	Adaptive solution strategies for coupled porous media flow / free flow systems	33
2.26	FEM-FEM and FEM-BEM Coupling within the Dune Computational Software Environment	34
2.27	Taking advantage seamlessly of new heterogeneous architectures, with a C++ DSEL to implement lowest-order methods for diffusive problem on general meshes	35
2.28	Hybrid Parallelization of Assembly in Dune	37
2.29	Parallel Geometric Multigrid in deal.II	38
2.30	Preparing DUNE for future HPC architectures	39

Chapter 1

Keynote lectures

1.1 Surviving and thriving as an open scientific software project

Garth N. Wells^a

Openness in scientific software is unfortunately still not the norm in computational science and engineering research. A lack of openness hinders scientific progress and is an enormous drain on resources. There are however many excellent examples of open, community-driven scientific software development which foster and make possible sharing, constructive development and research transparency. I will reflect on my own experiences of having used many open software packages in research, observing numerous open scientific software projects and having been involved in the FEniCS Project, with some thoughts on how to build, strengthen and promote open scientific software development.

^aHibbitt Reader in Solid Mechanics, Department of Engineering, University of Cambridge gnw20@cam.ac.uk

1.2 Large-Eddy Simulation with OpenFOAM: Practice, Applications and Future Challenges

Ville Vuorinen^a

Computational fluid dynamics (CFD) simulation of internal combustion engines requires high performance computing when modern simulation techniques relying on high space and time resolution is required. In my presentation, I summarize experiences gained at the Aalto University on the application of the open source code OpenFOAM for computational fluid dynamics (CFD) with regard to combustion and mixing problems occurring in internal combustion engines in simplified and realistic geometries.

First, I cover the practical side of code implementation where mathematical expressions, such as $\delta U = \Delta t(-\nabla \cdot U \otimes U) + \nu \Delta U$, occurring typically in fractional step methods, can be conveniently programmed as

```
dU = dt*(-fvc::div(U*U) + nu*fvc::laplacian(U));
```

Second, I will provide an overview of our recent progress in the simulation of highly compressible flow, moving mesh for in-cylinder flow, and combustion.

Last, I outline a few possible directions for future development of the code with regard to large-eddy simulation and direct numerical simulation.

Bibliography

- [1] Wehrfritz A., Vuorinen V., Kaario O., and Larmi M., *Large Eddy Simulation of High Velocity Fuel Sprays: Studying Mesh Resolution and Breakup Model Effects for Spray A*, Atomization and Sprays, **23**, 5, 419-442, (2013).
- [2] Vuorinen V., Larmi M., Scletter P., Fuchs L., and Boersma B., *A Low-Dissipative, Scale-Selective Discretization Scheme for the Navier-Stokes Equations*, Computers & Fluids, **70**, 30, 195-205, (2012).
- [3] Vuorinen V., Duwig C., Yu J., Tirunagari S., Kaario O., Larmi M., and Boersma B., *Large-Eddy Simulation of Highly Underexpanded Transient Gas Jets*, Phys.Fluids, **25**, 016101, (2013).
- [4] Vuorinen V., Wehrfritz A., Duwig C., and Boersma B., *On the Implementation of Low-Dissipative Runge-Kutta Projection Methods for Time Dependent Flows Using OpenFOAM*, Computers & Fluids, **93**, 10, 153-163, (2014).

^aAalto University School of Engineering, Department of Energy Technology ville.vuorinen@aalto.fi

1.3 Modeling waves with PyClaw: design, development, and applications

David Ketcheson ^a

PyClaw is an adaptation of the Clawpack code in which Python is used for all non-compute-intensive code components. It achieves serial performance similar to the original Clawpack code, but has a more convenient interface, runs on massively parallel distributed machines, and includes additional high-order accurate numerical algorithms. It also allows new applications to be rapidly prototyped in pure Python. I will describe how PyClaw development has been facilitated by the use of existing modern numerical libraries and open source tools. I will also present some recent applications of PyClaw in the fields of detonation waves, shallow water waves, and electromagnetism.

^aDivision of Computer, Electrical, and Mathematical Sciences and Engineering King Abdullah University of Science and Technology (KAUST) david.ketcheson@kaust.edu.sa

1.4 Firedrake and dolfin-adjoint: Defeating the complexity barrier in simulation generation by composing abstractions

David Ham^a

The creation of simulation software confronts an explosion of complexity: demands to simulate more complex systems require more complex numerics, and the effective use of ever-more complex hardware. Facing this challenge means effectively combining skills from across computer science, numerical analysis, and the scientific fields to which the simulation is to be applied. However the complexity of these fields means that it is not feasible for individual scientists to simultaneously be masters of them all. Progress requires that systems are developed which create sufficiently robust separations of concerns that experts in each field can make advances in their own research agendas while working on code which can be immediately employed by those in the other fields.

In this talk I will present two projects in which I am involved which are based on achieving this separation of concerns. In each case, the core idea is the composition of abstracted representations of computational problems. Firedrake is a toolchain for the solution of partial differential equations by the finite element method. It presents a user interface substantially identical to that of the well-known FEniCS project, but employs a different implementation of this interface based on the PyOP2 performance-portable unstructured mesh library, the COFFEE vectorising loop nest compiler, and PETSc. The result is a lightweight, responsive and fast implementation of this paradigm embodied in a very small number of lines of code.

Dolfin-adjoint operates at a still-higher level of abstraction to automate the solution of the tangent-linear and adjoint equations to a time varying finite element problem. For dolfin-adjoint, it is the capture of the high-level mathematical structure of the finite element problem at runtime which creates the required level of abstraction. This captured mathematical structure enables dolfin-adjoint to reason about and differentiate the forward model in a manner vastly more efficient and robust than the traditional algorithmic differentiation approach. The result is that adjoint calculations can be specified by adding a handful of lines of code to the original source, and that the resulting adjoint simulations execute with an efficiency approaching the theoretical optimum.

^aDepartment of Mathematics, Imperial College, London, david.ham@imperial.ac.uk

Chapter 2

Contributed lectures

2.1 HiFlow³ – A Hardware-Aware Parallel Finite Element Package

Teresa Beck^a, Simon Gawlok^b

HiFlow³ [1] is a multi-purpose finite element software providing powerful tools for the efficient and accurate solution of a wide range of problems modeled by partial differential equations (PDEs). Based on object-oriented concepts and the full capabilities of C++ the HiFlow³ project follows a modular and generic approach for building efficient parallel numerical solvers. It provides highly capable modules dealing with the mesh setup, finite element spaces, degrees of freedom, linear algebra routines, numerical solvers, and output data for visualization. Parallelism – as the basis for high performance simulations on modern computing systems – is introduced on two levels: coarse-grained parallelism by means of distributed grids and distributed data structures, and fine-grained parallelism by means of platform-optimized linear algebra back-ends.

The software package HiFlow³ offers relevant advantages to the numerical simulation of phenomena from a wide range of research topics: In the field of uncertainty quantification (UQ), HiFlow³ provides a software module for using Polynomial Chaos expansions to model uncertainties in physical problems by a linear algebra framework. Standard iterative solvers can be employed to solve the associated linear systems of equations. In addition, HiFlow³ supports mean based preconditioning to accelerate convergence for UQ problems. In the field of Computational Fluid Dynamics (CFD) and Meteorology, HiFlow³ enables the numerical simulation of various flow phenomena by providing highly scalable preconditioners and solvers for both linear and nonlinear systems. A grid- and time-adaptive simulation framework allow for a simulation of multiscale flow phenomena. HiFlow³ facilitates various a posteriori error estimators that can be used to control adaptivity in space and time.

We present HiFlow³ and outline its underlying modular concept. By means of multiple showcasing scenarios we present its applicability and feasibility to various fields of application, such as a goal-oriented adaptive simulation of tropical cyclones [2]. Further, we address its performance and scalability on HPC systems by considering a CFD benchmark problem [3].

Bibliography

- [1] V. Heuveline et. al, HiFlow3: A Hardware-Aware Parallel Finite Element Package, in *Tools for High Performance Computing 2011*, Springer (2012), 139–151.
- [2] M. Baumann, V. Heuveline, L. Scheck, S. C. Jones, Construction of Economical Meshes for Tropical Cyclone Forecasting, *30th Conference on Hurricanes and Tropical Meteorology* (2012).
- [3] V. Heuveline, E. Ketelaer, S. Ronnas, M. Schmidtobreick, M. Wlotzka, Scalability Study of HiFlow3 based on a Fluid Flow Channel Benchmark, *Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)* (2012).

^aEMCL, Uni Heidelberg, Teresa.Beck@iwr.uni-heidelberg.de

^bEMCL, Uni Heidelberg, Simon.Gawlok@iwr.uni-heidelberg.de

2.2 PyOP2: a performance-portable framework for parallel computations on unstructured meshes

Florian Rathgeber^a, Lawrence Mitchell^b, David Ham^c, Gheorghe-Teodor Bercea^d, Fabio Luporini^e, Paul H.J. Kelly^f

Many numerical algorithms and scientific computations on unstructured meshes can be viewed as the independent application of a local operation, often called a kernel, everywhere on a mesh, which lends itself naturally to parallel computation. Achieving good performance across a diverse range of contemporary parallel architectures however remains a challenge for modern scientific codes.

We present PyOP2, a high-level embedded domain-specific language (DSL) for the parallel execution of computational kernels on unstructured meshes described by sets of entities such as vertices, edges and cells, and the connectivity between those sets forming the topology of the mesh. PyOP2 generates problem-specific code for a range of hardware architectures, targeting multi-core CPUs, GPUs and accelerators and distributed parallel computations with MPI. Backend-specific code tailored to each specific problem is generated, just-in-time compiled and efficiently scheduled for parallel execution at runtime.

Optimised implementations for different hardware architectures can be automatically generated without any changes to a single high-level source, where the details of the parallel execution are abstracted away from the user. This makes PyOP2 suitable as an intermediate layer for building computationally efficient, maintainable, a composable scientific applications.

^aDepartment of Computing, Imperial College London, f.rathgeber@imperial.ac.uk

^bDepartment of Computing, Imperial College London, lawrence.mitchell@imperial.ac.uk

^cDepartment of Mathematics & Computing, Imperial College, david.ham@imperial.ac.uk

^dDepartment of Computing, Imperial College London, gheorghe-teodor.bercea08@imperial.ac.uk

^eDepartment of Computing, Imperial College London, f.luporini12@imperial.ac.uk

^fDepartment of Computing, Imperial College London, p.kelly@imperial.ac.uk

2.3 Recent Development in NGSolve

Joachim Schöberl ^a

We give an overview of our open-source software-project Netgen/NGSolve. Its strength are high order scalar and vectorial finite elements as well as hybrid discontinuous Galerkin methods, and fast solvers. We discuss the representation of the partial differential equation by high-level data-structures. We also show some details on recent development on utilizing SIMD and GPU architectures.

^aInstitute for Analysis and Scientific Computing, Vienna University of Technology
joachim.schoeberl@tuwien.ac.at

2.4 Experiences using 2decomp&fft to solve Partial Differential Equations using Fourier Spectral Methods

Sudarshan Balakrishnan^a , Abdullah Bargash^b , Gong Chen^c , Brandon Cloutier^d ,
Ning Li^e , Dave Malicke^f , **Benson K. Muite**^g , Michael Quell^h , Paul Riggeⁱ ,
Mamdouh Solimani^j , Andre Souza^k , Mark Van Moer^l , Jeremy West^m

We describe experiences using the library 2decomp&fft[1] to solve partial differential equations on parallel computers. An overview of methods and platforms used as well as problems encountered will be given. Example programs and documentation can be found in [2].

Bibliography

- [1] N. Li, S. Laizet, 2DECOMP&FFT – A highly scalable 2D decomposition library and FFT interface, *Cray User Group 2010 conference*, Edinburgh, 2010. <http://www.2decomp.org/>
- [2] S. Balakrishnan, A.H. Bargash, G. Chen, B. Cloutier, N. Li, B.K. Muite, M. Quell, P. Rigge, M. Solimani, A. Souza, A.S. Thiban, J. West, D. Malicke, M. Van Moer, http://en.wikibooks.org/wiki/Parallel_Spectral_Numerical_Methods

^aUniversity of Michigan, sudarsh@umich.edu

^bKing Abdulaziz University, un-i2011@hotmail.com

^cUniversity of Chicago, gc@math.uchicago.edu

^dUniversity of Michigan, brandon.clouter@gmail.com

^eNumerical Algorithms Group, ning.li@nag.co.uk

^fUniversity of Michigan, dmalicke@umich.edu

^gUniversity of Tartu, benson.muite@ut.ee

^hTechnical University of Vienna, michael.quell@yahoo.de

ⁱUniversity of California at Berkley, sirmichigan@gmail.com

^jKing Abdullah University, m80ismail@gmail.com

^kUniversity of Michigan, sandre@umich.edu

^lUniversity of Illinois at Urbana Champaign, mvanmoer@iillinois.edu

^mUniversity of Michigan, jeremyw@umich.edu

2.5 Introduction to hpGEM: Software library for discontinuous Galerkin methods

Anthony R. Thornton^a, *et. al.*^b

We present a fast and user friendly open-source software framework for discontinuous Galerkin (DG) finite element methods (FEM), hpGEM [1]. Data structures and methods provided through this environment are common for many (discontinuous) FEM problems. As a result, hpGEM facilitates: fast implementation of finite element discretisations (including space-time discretisations), the assessment of algorithms, and their applications to real-world problems. Its design is focused on two essential things:

1. easy-to-use interfaces;
2. fast methods in combination with robust data structures.

User friendly interfaces are provided via simple wrapper-functions, which enable access to powerful methods and data structures without a need to understand a complex architecture of the packages highly object oriented C++ kernel. The wrappers introduce an opportunity to implement a DG discretisation on a higher level of abstraction via several command-calls.

Computational efficiency is achieved via robust data structures and fast mathematical algorithms. The package enables a possibility of multigrid support, as well as h- and p-adaptivity. The feasibility of the approach is demonstrated via an implementation of several applications ranging from standard Laplace problems, to Maxwell's equations and also a novel discretisation of Hamiltonian structure for the linearised (in)compressible fluid flow in a three-dimensional rotating domain.

If you would like more information about the code, this can be found at the hpGEM website <http://einder.ewi.utwente.nl/hpGEM/>.

Bibliography

- [1] L. Pesch, A. Bell, W. E. H. Sollie, V. R. Ambati, O. Bokhove, and J. J. W. Vegt. hpgem a software framework for discontinuous galerkin finite element methods. *ACM Transactions on Mathematical Software*, 33(4), 2007.

^aUniversity of Twente, a.r.thornton@utwente.nl

^bhpGEM has been developed over a number of years by a range of people. A full list of contributors can be found at <http://einder.ewi.utwente.nl/hpGEM/about-the-code/team>

2.6 The BEM++ boundary element library

Elwin van 't Wout ^a

The BEM++ boundary element [1] library is an open source library for the development of a range of boundary element applications. It is developed as a collaboration between University College London, the University of Reading, and the University of Warwick. Its core is a templated C++ library that offers Galerkin discretisation of integral equations for Laplace, Helmholtz, and Maxwell problems. It interfaces with various external libraries, such as DUNE for grid management and Trilinos for fast iterative solvers. The library can be used directly from C++ or via a comprehensive Python interface. In this talk we discuss the design of BEM++ and show various application examples, including medical imaging and time-domain problems.

Bibliography

- [1] W. Smigaj, T. Betcke, S. Arridge, J. Phillips, M. Schweiger, *Solving Boundary Integral Problems with BEM++*, ACM Trans. Math. Softw., to appear

^aDepartment of Mathematics, University College London elwinwout@live.nl

2.7 DOpElib: The differential equations and optimization environment

C. Goll^a, T. Wick^b, W. Wollner^c

In this talk, we describe the *Differential Equations and Optimization Environment* (DOpElib). DOpElib is a software library that provides a unified interface to high level algorithms such as time-stepping methods, nonlinear solvers and optimization routines. This structure ensures that, first of all, the user is only required to write those sections of code that are specific to the considered problem. Secondly, the exchange of parts of the used routines is possible with only a few lines of code to change instead of large reimplementations.

The current implementation provides an interface to the finite element library `deal.II` as a basis for the solution of the considered partial differential equation.

The talk illustrates the design principles and various features of DOpElib and provides some numerical results as demonstration for the versatility of the software.

^aUniversität Heidelberg, christian.goll@iwr.uni-heidelberg.de

^bUT Austin, twick@ices.utexas.edu

^cUniversität Hamburg, winnifried.wollner@uni-hamburg.de

2.8 Extruded Meshes for High Aspect Ratio Simulations in Firedrake and PyOP2

Gheorghe-Teodor Bercea^a, Andrew T. T. McRae^b, Florian Rathgeber^c, Fabio Luporini^d, Nicolas Lorient^e, David A. Ham^f, Paul H. J. Kelly^g, **Lawrence Mitchell^h**

We present the extension of Firedrake to support extruded meshes of prismatic elements suitable for simulation in high aspect ratio domains. Important motivating examples of such domains include the atmosphere and ocean, in which the depth is orders of magnitude smaller than the horizontal extent. We call the underlying meshes for this type of problems extruded meshes. By using modified versions of the Unified Form Language (UFL), the FEniCS[1] Form Compiler (FFC) and PyOP2, Firedrake is able to automatically generate and execute the computational kernels over extruded domains. In Firedrake, a 3D extruded mesh consists of an unstructured 2D (*horizontal*) base mesh and a number of layers (*vertical*). Every layer of elements will have the topology of the base mesh making the extrusion direction structured. We will show how the structured accesses along the vertical structure can be used to offset the performance cost of using an unstructured mesh in the horizontal. Firedrake also provides a flexible way of building function spaces as a tensor product of vertical and horizontal finite elements. In this talk we will give evidence of the performance gains behind using extruded meshes and also attempt to offer some insight into the space of problems that are suited to geometrically semi-structured discretisations. We will present a comparison of the peak performance achievable in two use cases: the Helmholtz and Burgers equations. The results obtained for very simple problems on extruded meshes suggest that as much as 80% of the achievable peak machine performance can be reached. We will use this as a starting point for presenting a systematic study of the impact of different real-world problem formulations on the achievable performance.

Bibliography

- [1] A. Logg, K.-A. Mardal, G. N. Wells et al. Automated Solution of Differential Equations by the Finite Element Method, Springer, (2012).

^aDepartment of Computing, Imperial College London, gheorghe-teodor.bercea08@imperial.ac.uk

^bDepartment of Mathematics, Imperial College London and The Grantham Institute for Climate Change
a.mcrae12@imperial.ac.uk

^cImperial College London, f.rathgeber10@imperial.ac.uk

^dDepartment of Computing, Imperial College London, f.luporini12@imperial.ac.uk

^eDepartment of Computing, Imperial College London, n.lorient@imperial.ac.uk

^fDepartment of Mathematics, Imperial College London and The Grantham Institute for Climate Change
david.ham@imperial.ac.uk

^gDepartment of Computing, Imperial College London, p.kelly@imperial.ac.uk

^hDepartment of Computing, Imperial College London, lawrence.mitchell@imperial.ac.uk

2.9 Automated Build System for PDE Toolboxes

Uwe Köcher^a, Markus Bause^b

Nowadays there exists a zoo of modern high-performance PDE toolboxes. As stated in the announcement of this workshop, this power does not come for free and thus, those state of the art PDE toolboxes, e.g. deal.II, FEniCS, DUNE and others, consists usually of a bunch of software packages which are finally linked together. A first build of a toolchain can be very frustrating for beginners and non-experienced users. Even for experienced users, preparing the toolbox can be very time consuming since different platforms, e.g. the personal workstation and notebook, compute node or HPC-cluster, and different operating systems may have different requirements on the builds.

To overwhelm this problem, the FEniCS-authors implemented *dorsal* under the Lesser GPL v3, a bash-script based solution for the automated build of their toolbox on different platforms, i.e. Debian, Ubuntu, Red Hat, Macintosh, and more. In addition, they provide initial support for other software packages as for deal.II 7.x and DUNE. Dorsal is optimised for building FEniCS on Debian based platforms and since that the provided build support for other toolboxes may be out of date.

In this contribution, we present *candi* (compile-and-install) as an essential extended fork from dorsal, allowing flexible build and install for deal.II 8.x and developers version and FEniCS on Red Hat based platforms. More precisely, for deal.II it allows various types of builds from a minimal to a fully customised version using MPI, Trilinos, PETSc, and other supported libraries. Additionally, it is possible to install different types of the same library on the same machine since candi automatically creates modulefiles for the specific build, which allows a “hot-plug”-change of the used toolbox; cf. for details on candi [1].

Bibliography

- [1] U. Köcher: Variational space-time methods for the elastic wave equation: Applied and Implemented for the Simulation of Structural Health Monitoring Systems, Doctoral thesis, in preparation, 2014.

^aHelmut-Schmidt-University, University of the German Federal Armed Forces Hamburg, koecher@hsu-hamburg.de

^bHelmut-Schmidt-University, University of the German Federal Armed Forces Hamburg, bause@hsu-hamburg.de

2.10 Entropy-based viscous regularization for the multi-dimensional Euler equations in low-Mach regimes

Marc O. Delchini^a , Jean C. Ragusa^b

The entropy viscosity method, introduced by Guermond et al. [1, 2], is extended to the multi-dimensional Euler equations for both subsonic (very low Mach numbers) and supersonic flows. We show that the current definition of the viscosity coefficients [1] is not adapted to low-Mach flows and we provide a robust alternate definition valid for any Mach number value. The new definitions are derived from a low-Mach asymptotic study. In addition, the entropy minimum principle is used to derive the viscous regularization terms for Euler equations with variable area for nozzle flow problems. Various 1- and 2-D numerical tests are presented : flow in a convergent-divergent nozzle, Leblanc shock tube, subsonic flow around a 2-D cylinder and over a circular hump, and supersonic flow in a compression corner. Convergence studies are performed using analytical solutions in 1-D. Both the ideal gas and stiffened gas equations of state are employed.

Bibliography

- [1] J. L. Guermond, R. Pasquetti, Entropy viscosity method for nonlinear conservation laws, *Journal of Comput. Phys* 230 (2011) 4248–4267.
- [2] J. L. Guermond, R. Pasquetti, Entropy viscosity method for high-order approximations of conservation laws, *Lecture Notes in Computational Science and Engineering* 76 (2011) 411–418.

^aDepartment of Nuclear Engineering, Texas A&M University, College Station, TX 77843, USA, delchmo@tamu.edu

^bDepartment of Nuclear Engineering, Texas A&M University, College Station, TX 77843, USA, jean.ragusa@tamu.edu

2.11 pyMOR – Model Order Reduction with Python

Stephan Rave^a, René Milk^b

In this contribution we present pyMOR [1], a new open source library for building model order reduction applications with the Python programming language. Its main focus lies on the easy application of reduced basis methods ([2] and references therein) to parameterized partial differential equations solved by external high-dimensional discretization packages. At the same time, pyMOR provides a built-in discretization toolkit based on the NumPy/SciPy [3] software stack, which makes it an attractive tool for teaching and rapid exploration of new model reduction algorithms.

We give a brief overview of pyMOR's main components and explain in detail the design philosophy and choices that allow us to arrive at a set of completely generic reduction algorithms which can be readily incorporated with any external solver connected to pyMOR via its lightweight operator and vector interfaces. We discuss how this approach simultaneously empowers the solver with both the flexibility and interactivity of a modern dynamic language like Python. We conclude our presentation with a practical example of the provided infrastructure to integrate existing DUNE [4] discretizations with pyMOR.

Bibliography

- [1] R. Milk, S. Rave, F. Schindler, pyMOR - Model Order Reduction with Python, <http://pymor.org>.
- [2] B. Haasdonk, M. Ohlberger, Reduced basis method for finite volume approximations of parametrized linear evolution equations, *M2AN Math. Model. Numer. Anal.*, **42** (2008), 277–302.
- [3] T. Oliphant, Python for Scientific Computing, *Computing in Science & Engineering*, **9** (2007), 10–20.
- [4] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, O. Sander, A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part I: Abstract Framework, *Computing*, **82** (2008), 103–119.

^aInstitute for Numerical and Applied Mathematics, University of Münster, stephan.rave@uni-muenster.de

^bInstitute for Numerical and Applied Mathematics, University of Münster, rene.milk@uni-muenster.de

2.12 Interactive Simulations using Parallel and Event-Driven Localized Reduced Basis

Andreas Buhr^a, Christian Engwer^a, Mario Ohlberger^a

Interactive simulation tools reacting on user input at any time have to be designed in a different way than 'classical' simulation programs. For interactive applications, event-driven architectures are the natural choice. With HPX [1], there exists a framework for event driven parallel computations on large clusters, hiding network latencies. It allows for a complete separation of parallelization code and user code. The same code can run on a single computer or in clouds around the globe. These concepts are attractive for numerical codes.

We will present a runtime system for interactive simulations, following the concepts of HPX. Making use of localized Reduced Basis methods we minimize recomputation and communication after geometry changes, thus leading to fast response on user interaction. Geometry processing, meshing, matrix assembly and reduced basis generation are realized using DUNE [2] and pyMOR [3] in a localized, event-driven fashion. All intermediate quantities in calculations are versioned, sophisticated dependency management triggers recomputation as necessary.

Within this framework, we present the implementation of different Multiscale and Reduced Basis methods. Parallel, interactive simulations of electromagnetic fields in 2D geometries are shown.

Bibliography

- [1] H. Kaiser, M. Brodowicz and T. Sterling: ParalleX: An Advanced Parallel Execution Model for Scaling-Impaired Applications, *International Conference on Parallel Processing Workshops* 2009
- [2] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, R. Kornhuber, M. Ohlberger, O. Sander. A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part II: Implementation and Tests in DUNE. *Computing*, 82(2-3), 2008, pp. 121-138
- [3] pyMOR - Model Order Reduction with Python, <http://pymor.org>

^aInstitute for Numerical and Applied Mathematics, Münster
{andreas.buhr, christian.engwer, mario.ohlberger}@uni-muenster.de

2.13 Abstraction of Multiscale Methods: Theory and Implementation

Sven Kaulmann^a, René Milk^b, Mario Ohlberger^b

Among numerous other applications, multiscale problems as, for example, arising in the simulation of complex flows in reservoir engineering, promise good applicability of exa-scale computing techniques. In our contribution, we will introduce a mathematical abstraction of multiscale methods [7] such as the Multiscale Finite Element method [5], the Heterogeneous Multiscale method [4] and the Variational Multiscale method [6]. Based on this unified mathematical abstraction layer, we introduce a hybrid parallelization approach that reflects the different layers of these multiscale methods. Our implementation combines techniques such as Shared- and Distributed Memory parallelization and GPU/MIC accelerators using the Distributed and Unified Numerics Environment DUNE [1, 3, 2] to achieve scalability and efficiency on current and future, potentially highly heterogeneous, peta- and exa-scale computing clusters. We conclude our presentation with preliminary numerical results for the Multiscale Finite Element method.

Bibliography

- [1] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöforn, Ralf Kornhuber, Mario Ohlberger, and Oliver Sander. A generic grid interface for parallel and adaptive scientific computing. Part II: Implementation and Tests in DUNE. *Computing*, 82(2-3):121–138, June 2008.
- [2] Andreas Dedner, Bernd Flemisch, and Robert Klöforn, editors. *Advances in DUNE: Proceedings of the DUNE User Meeting, Held in October 6th-8th 2010 in Stuttgart, Germany*. Springer, 2012 edition, April 2012.
- [3] Andreas Dedner, Robert Klöforn, Martin Nolte, and Mario Ohlberger. A generic interface for parallel and adaptive discretization schemes: abstraction principles and the Dune-Fem module. *Computing*, 90(3-4):165–196, August 2010.
- [4] Weinan E and Bjorn Engquist. The heterogeneous multiscale methods. *Communications in Mathematical Sciences*, 1(1):87–132, 2003.
- [5] Thomas Y Hou. A Multiscale Finite Element Method for Elliptic Problems in Composite Materials and Porous Media. *Journal of Computational Physics*, 134(1):21–21, May 1997.
- [6] Thomas J R Hughes. Multiscale Phenomena - Greens-Functions, the Dirichlet-to-Neumann Formulation, Subgrid Scale Models, Bubbles and the Origins of Stabilized Methods. *Computer Methods in Applied Mechanics and Engineering*, 127:387–401.
- [7] Mario Ohlberger. Error control based model reduction for multiscale problems. In Angela Handlovičová, Zuzana Minarechová, and Daniel Ševčovič, editors, *Algoritmy 2012*, pages 1–10. Slovak University of Technology in Bratislava, April 2012.

^aInstitute for Numerical and Applied Mathematics, University of Münster and Institute for Applied Analysis and Numerical Simulation, University of Stuttgart

^bInstitute for Numerical and Applied Mathematics, University of Münster

2.14 An Adaptive Framework for a Heterogeneous Multiscale Method

Matthias Maier^a

A large class of modeling problems in Physics and Engineering is of multiscale character, meaning, that relevant physical processes act on highly different length scales. This usually implies high computational cost for a full resolution of the problem. One way to avoid such a full resolution are multiscale schemes, where, generally speaking, an effective model is solved on a coarse scale with upscaled, effective parameters that are determined with the help of localized sampling problems on a fine scale.

Multiscale schemes introduce significant complexity with respect to sources of error, not only are there discretization errors on a coarse and fine scale, but also a model error introduced by the modeling assumption. This makes suitable a posteriori strategies highly necessary.

We present a modification of the well known Heterogeneous Multiscale Method [1] for elliptic advection-diffusion problems together with an adaptive framework based on a posteriori error estimation with the help of the Dual-Weighted-Residual Method [2]. Quantitative a posteriori error estimators are derived that allow for different adaptive strategies.

Finally, efficient data structures and algorithms, as well as its implementation in the DEAL.II library [3] are discussed.

Bibliography

- [1] W. E, B. Engquist, The Heterogeneous Multiscale Methods, *Communications in Mathematical Sciences* **1** (2003), 87–132.
- [2] R. Becker, R. Rannacher, An Optimal Control Approach to A Posteriori Error Estimation in Finite Element Methods, *Acta Numerica* **10** (2001), 1-102
- [3] W. Bangerth, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, T.D. Young, The DEAL.II Library Version 8.0 (2013), online available at: <http://arxiv.org/abs/1312.2266>

^aInstitute of Applied Mathematics, Heidelberg University, matthias.maier@iwr.uni-heidelberg.de

2.15 Solving PDEs in High Space Dimension using DUNE

Martin Nolte^a

Partial differential equations in high space dimensions arise quite naturally in mathematical modelling. In optimal control, for example, Bellman's value function, mapping each initial state to the minimal cost for achieving a desired goal, solves a nonlinear Hamilton-Jacobi equation. Depending on the observed process, the state space can easily be of arbitrary dimension.

The development of simulation software solving high-dimensional PDEs even for simple problems is a challenging task of its own. Since human imagination is limited to the three-dimensional space surrounding us, we have to rely on purely mathematical concepts. The abstract mathematical definitions and generic interfaces of DUNE form a sound basis for the solution of high-dimensional PDEs.

From a practical point of view, discretization meshes contain a vast number of elements and parallelization is inevitable. Still, careful programming is required not to exhaust application memory due to redundant information or over-aggressive caching.

This talk will illustrate challenges in solving high-dimensional PDEs and their solutions available in DUNE and DUNE-FEM.

^aUniversity of Freiburg, nolte@mathematik.uni-freiburg.de

2.16 LifeV: parallel framework and applications to blood-flow simulations

Simone Deparis^a, Gwenol Grandperrin, Radu Popescu, Cristiano Malossi

LifeV, <http://www.lifev.org>, is an object oriented Finite Element Library for the approximation of partial differential equations. It is distributed under LGPL by CMCS - EPFL, E(CM)2 - Emory, MOX - Polimi, and REO, ESTIME - INRIA. LifeV is a research code oriented to the development and test of new numerical methods and algorithms, whose aim it to be an effective tool for solving complex “real-life” engineering problems.

The parallelism in LifeV is based on MPI and Trilinos, <http://www.trilinos.org>. The parallelization of the finite element loop is achieved through domain decomposition (DD), in particular mesh partitioning with Zoltan or Parmetis, preconditioners build via Ifpack or ML, but specialized for the underlying physical problem. Recent developments investigated a hybrid paradigm, where the number of processors and the subdomain are different. Each subdomain of the DD framework is assigned to a subgroup of MPI processes or, alternatively, to several cores in a openMP environment. The subdomain solve is done by a parallel direct solve or an approximate inverse.

In the second part of the talk we focus on a geometrical multiscale model for the cardiovascular tree. This allows to consider different levels of complexity and it accounts for both the systemic circulation and local three-dimensional features. Three dimensional models which includes fluid-structure interaction (FSI) are needed to understand the local behavior of the flow, while to roughly reproduce the global arterial system it is possible to use a network of one dimensional hyperbolic models.

^aChair of Modeling and Scientific Computing, MATHICSE - EPFL, Station 8, 1015 Lausanne, Switzerland, simone.deparis@epfl.ch

2.17 Domain decomposition based fluid-structure interaction algorithm based on Steklov-Poincaré operators

Davide Forti^a, Simone Deparis*

In the last two decades the interest for Fluid-Structure Interaction (FSI) problem has increased significantly and simulations have become more feasible thanks to the advances in computer technology and numerical methods. Typically, two main algorithms are considered for the numerical simulation of the coupled fluid–structure problem: monolithic approaches [1] wherein the equations governing the flow and the displacement of the structure are solved simultaneously with a single solver, and partitioned ones, where the equations governing the flow and the displacement of the structure are solved separately, by means of two distinct solvers. In this framework we present a Steklov-Poincaré algorithm [2] for the numerical simulation of fluid-structure interaction problems that allows for reducing the whole FSI system to an equation involving only interface variables. We will compare monolithic and Steklov-Poincaré algorithms for FSI from the point of view of performances, code reusability and their parallel implementation, addressing both the cases of conforming and nonconforming meshes at the fluid-structure interface. The latter makes use of radial basis functions (RBF) interpolation. We present a variant of RBF which is local, exact for constant functions, and scalable [3]. The algorithms have been implemented in the finite element library LifeV parallel (www.lifev.org).

Bibliography

- [1] P. Crosetto, S. Deparis, G. Fourestey, and A. Quarteroni. Parallel Algorithms for Fluid-Structure Interaction Problems in Haemodynamics. *SIAM Journal on Scientific Computing*, 33(4):1598–1622, 2011.
- [2] S. Deparis, M. Discacciati, G. Fourestey, and A. Quarteroni. Fluid-structure algorithms based on Steklov-Poincaré operators. *Computer Methods in Applied Mechanics and Engineering*, 195:5797–5812, 2006.
- [3] S. Deparis, D. Forti, and A. Quarteroni. A rescaled localized radial basis functions interpolation on non-cartesian and non-conforming grids. Submitted, 2014.

^aChair of Modeling and Scientific Computing, MATHICSE - EPFL, Station 8, 1015 Lausanne, Switzerland, davide.forti@epfl.ch, simone.deparis@epfl.ch

2.18 HiFlow³-based Numerical Simulations for Cognition–Guided Surgery

Jonas Kratzke^a , Nicolai Schoch^b

The research in the field of medical engineering done at the Engineering Mathematics and Computing Lab (EMCL) is part of a common research effort of the collaborative research center (SFB TRR 125) 'Cognition-Guided Surgery'. It stands for the idea of surgery, which lets itself be lead by machine cognition, i.e., by a technical, cognitive system, which not only executes programmed tasks, but also interprets a given situation and acts accordingly, therein supporting the surgeon.

As part of this vision, medical simulations play an important role. They not only can contribute to the overall understanding of human physiology, but also enhance diagnosis tools, give additional information for risk analysis and hence allow for optimal surgery planning.

Due to the medical application requirements (high accuracy and real-time simulations), our simulations are implemented in HiFlow³[1], and optimized with respect to hardware-aware numerics. Moreover, when making surgical operation decisions, analysing risks and specifying clinical treatment scenarios on the basis of the simulation results, we consider the simulation quality using methods of Uncertainty Quantification (UQ) in order to guarantee reliable information. We present several medical application scenarios as arising from liver and heart surgery and from the context of aortic blood flow analysis.

Driven by the need for integration of our software into the clinical operation treatment workflow, we work on a corresponding software interface in the context of the Medical Simulation Markup Language (MSML) project[2]. The corresponding workflow starts from image acquisition, goes via image processing, model setup and simulation, and ends with post-processing tasks, such as visualization and clinical statistics.

A crucial point in the development of medical simulations is the issue of a validation of the simulation methods. In this context, we present our ongoing activities in the hand-in-hand development of an aortic silicon phantom and computer model, as well as our efforts on the U.S. FDA's benchmarking call concerning the simulation of an idealized ventricular assist device.

Bibliography

- [1] V. Heuveline et. al, HiFlow³ – A Hardware-Aware Parallel Finite Element Package, in *Tools for High Performance Computing 2011*, Springer (2012), 139–151.
- [2] S. Suwelack et. al, The Medical Simulation Markup Language – Simplifying the Biomechanical Modeling Workflow, in *Medicine Meets Virtual Reality*, MMVR21 (2014).

^aEMCL, Uni Heidelberg, jonas.kratzke@iwr.uni-heidelberg.de

^bEMCL, Uni Heidelberg, nicolai.schoch@iwr.uni-heidelberg.de

2.19 IsoGlib: an Isogeometric Analysis library for the solution of high-order Partial Differential Equations on surfaces

Andrea Bartezzaghi^a, Luca Dedè^a, Alfio Quarteroni^{ab}

In this work we consider the numerical approximation of high-order Partial Differential Equations (PDEs) on lower dimensional manifolds, specifically on surfaces in three dimensional spaces; as far as the spatial approximation is concerned, we use NURBS-based Isogeometric Analysis (IGA) [1] in the framework of the Galerkin method. This leads both to an exact geometrical description of the manifold when it is represented by B-splines or NURBS, as e.g. for conic sections, and to the ability of using global smooth basis functions, namely globally C^k -continuous, with $k \geq 0$. IGA presents different challenges from the point of view of the implementation with respect to standard Finite Elements libraries. In this work, we discuss the strategies adopted for the development and implementation of IsoGlib, a general IGA library for the numerical approximation of PDEs. In particular, we address the issue of solving high-order PDEs on surfaces, with and without periodic boundary conditions. We present some theoretical results for open and closed surfaces [2] [3] and some phase field applications on surfaces, as those described by the fourth-order Cahn-Hilliard equations [4] and the sixth-order crystal equation [5].

Bibliography

- [1] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*, John Wiley & Sons, 2009
- [2] L. Dedè, A. Quarteroni, Isogeometric Analysis for second order Partial Differential Equations on surfaces, *MOX-Report No. 06/2013*, 2013
- [3] A. Tagliabue, L. Dedè, A. Quarteroni, Isogeometric Analysis and Error Estimates for High Order Partial Differential Equations in Fluid Dynamics, *MOX-Report No. 27/2013*, 2013
- [4] J. Liu, L. Dedè, J. A. Evans, M. J. Borden, T. J. R. Hughes, Isogeometric Analysis of the advective Cahn-Hilliard equation: Spinodal decomposition under shear flow, *Journal of Computational Physics* 242:321-350, 2013
- [5] H. Gomez, X. Nogueira, An unconditionally energy-stable method for the phase-field crystal equation, *Computer Methods in Applied Mechanics and Engineering* 249-252:52-61, 2012

^aChair of Modeling and Scientific Computing (CMCS), MATHICSE - SB, École Polytechnique Fédérale de Lausanne, Station 8, 1015 Lausanne, Switzerland, andrea.bartezzaghi@epfl.ch

^bMOX - Modeling and Scientific Computing, Dipartimento di Matematica "F. Brioschi", Politecnico di Milano, via Bonardi 9, 20133 Milano, Italy (on leave)

2.20 TEMPO - a one step alternative to SUNDIALS

Philipp Birken^a

We present a C++ library for the solution of initial value problems, aimed at time dependent PDE systems. The library features time adaptive one step methods like DIRK, ESDIRK and Rosenbrock and uses a Jacobian-free Newton-Krylov (JFNK) method to solve the appearing systems [1]. Thus, it is an alternative to the well known software SUNDIALS [2]. However, replacing the BDF methods used in SUNDIALS with one step methods turns out to be more efficient for engineering accuracies. Furthermore, IMEX methods have been incorporated.

The library needs a function pointer to define the initial value problem and, due to the Jacobian-free construction, this is sufficient to solve the nonlinear systems. The basic data structure is a vector, thus when a data structure in the PDE solver based on points or other geometric properties is employed, this needs to be transformed. In case of preconditioning, a sparse block matrix format is provided with appropriate interfaces to define the preconditioner.

Bibliography

- [1] P. Birken, *Numerical Methods for the Unsteady Compressible Navier-Stokes Equations*, Habilitation Thesis, University of Kassel, 2012.
- [2] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, C. S. Woodward, *SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers*, ACM TOMS 31(3), 2005, pp. 363–396.

^aLund University, Centre for the Mathematical Sciences, Box 118, 22100 Lund, Sweden, philipp.birken@na.lu.se

2.21 PetIGA: High-Performance Isogeometric Analysis

A.M.A. Cortes^a, A. S. Rodriguez^a, D. A. G. Lozano^a, L.D. Dalcin^a, N.O. Collier^b, P. Vignal^a, V.M. Calo^a

PetIGA [1] is a high-performance library, built on top of PETSc, for NURBS-based isogeometric analysis discretization. By exploiting the tensor product nature of the basis functions, we are able to use the parallel data structures of PETSc, called DMDA's. These DA's build all the equation mappings needed for the parallel assembly of the linear system, and take care of the communication patterns for the user. Using this framework has allowed us to develop solvers for a variety of problems. The classical linear problems: Poisson, elasticity, Helmholtz, thin shells, advection-diffusion, and diffusion-reaction, but also the more challenging nonlinear time dependent problems: Cahn-Hilliard, Navier-Stokes-Korteweg, Variational Multiscale for Navier-Stokes, Phase-Field Crystal (PFC), etc.

Our recent developing effort is extending PetIGA to add the recently introduced [2] divergence-conforming B-spline spaces. Such spaces are suitable to discretize incompressible viscous flows, since it guarantees a point-wise divergence-free velocity field. We are also adding multi-fields capabilities, such that, these new velocity-pressure pair could be coupled with other physics, being defined by scalar or vector quantities.

Our goal is to present the PetIGA framework with some examples, and also shows parts its extension, solving Stokes and Navier-Stokes equations with divergence-conforming B-spline spaces.

Bibliography

- [1] N. Collier, L. Dalcin, and V.M. Calo, "PetIGA: High-Performance Isogeometric Analysis", submitted, 2013.
- [2] J. A. Evans and T. J. R. Hughes, "Isogeometric divergence-conforming B-splines for the Darcy-Stokes-Brinkman equations", *Mathematical Models and Methods in Applied Sciences*, 2013.

^aKAUST adriano.cortes@kaust.edu.sa

^bOak Ridge National Laboratory

2.22 Locally adaptive space-time grid in electromagnetic plasma simulations

Nina Elkina^a

The numerical simulation of relativistic plasmas by means of the solution the Vlasov-Maxwell equations is a tough problem simply because its high dimensionality (3 space + 3 velocity dimensions). To catch some physical meaning the problem requires at least one space 1d plus 3v velocity dimensions to track the distribution function evolving in time. Finally, in order to make reliable prediction for the experiment we need at least $2d+3v = 5$ dimensional codes. Under such conditions the ability to adapt the space and time grid resolution becomes very desirable feature to reduce an enormous computational loading. However, it has been long recognized that nonuniform mesh may cause spurious reflections of electromagnetic waves at abrupt change of grid resolution [1]. This effect is especially pronounced in finite-difference explicit schemes. The reason for such an unforgettable behaviour is that the dispersion relations of a variety of schemes fails to support physically correct wave propagation in particular in high frequency range. It is worth noting that the spurious reflections arise not only in electromagnetism but also well known for geophysical applications and can be found in other types of the wave equation. As a cure one can find a schemes with monotonic dispersion relation which always preserve the correct sign of the group velocity [2]. However, from practitioner's point of view the schemes with the monotonic dispersion relation have one essential problem of being at least partially implicit. As a result, the computations appears globally coupled and then require to solve the huge matrix equations or to perform extensive iterations.

The question that will be addressed in my talk is how to deal with the spurious effects on adaptive grid arising due to non-ideal numerical dispersion properties in application to the computational electromagnetics. This is related to the more general question of how to couple the adaptive grid Maxwell solver with nonlinear medium represented by the relativistic plasmas which also needs adaptation. In this talk, I will present the results of our investigation in both explicit and implicit schemes for the Maxwell equations and show how analysis of these models together with plasma helps us to gain some physical insight about nonlinear behaviour inside highly localized electron-positron plasma drops created by quantum cascades driven by the ultra-intense laser fields [3].

Bibliography

- [1] R. Vichnitsky, *Mathematics and Computers in Simulation*, pp. 344–353, 1981.
- [2] J. Frank, S. Reich, Tech. Rep. MAS-E0608, Centrum voor Wiskunde en Informatica, Amsterdam, 2004.
- [3] N. V. Elkina, A. M. Fedotov, I. Y. Kostyukov, M. V. Legkov, N. B. Narozhny, E. N. Nerush, H. Ruhl, *Physical Review Special Topics Accelerators and Beams* **14**, 2011, arXiv:1010.4528.

^aLudwig-Maximilians Universität, München, 80539, Germany, Nina.Elkina@physik.uni-muenchen.de

2.23 A computational approach for problems involving moving interfaces

Stefan Frei^a, Thomas Richter^b

Interface problems arise in a variety of applications, e.g. in multiphase flow and fluid-structure interaction problems. These two problem classes have in common that the solution is continuous, its derivatives however may be discontinuous across interface curves within the domain.

We present a computational approach [1] based on a modified finite element method that is able to approximate such interface problems with high accuracy. The proposed discretization is based on a local modification of the finite element basis and corresponds to a fitted finite element method. However, instead of moving mesh nodes, we resolve the interface locally by an adapted parametric approach. In contrast to alternative approaches like the Extended Finite Element (XFEM), we do not add additional degrees of freedom and hence structure and connectivity of the system matrix remain unchanged. As all modifications are applied locally, our scheme is very easy to implement and is well suited for time-dependent moving interface problems.

Bibliography

- [1] S. Frei, T. Richter: A locally modified parametric finite element method for interface problems, accepted for publication in: *SIAM Journal on Numerical Analysis* (2014)

^aInstitute of Applied Mathematics, stefan.frei@iwr.uni-heidelberg.de

^bInstitute of Applied Mathematics, thomas.richter@iwr.uni-heidelberg.de

2.24 A New Interface-based Approach to Heterogeneous Grid Coupling in DUNE

Christoph Gersbacher^a

Heterogeneous domain decomposition methods arise in the numerical approximation of physical phenomena when different kinds of (initial) boundary value problems have to be considered [4]. Their implementation is challenging due to the multitude of (possibly non-matching) grids, solvers, and coupling conditions to be taken into account. This is particularly true of problems that involve domains of different dimensions. Interface-based approaches to creating and handling coupled grid configurations are a successful attempt to reduce the complexity of multidomain applications [1, 3].

When coupling different grids their connectivity is usually established by common element overlap regions or interfaces, which are geometric notions. However, in some dimensionally heterogeneous problem settings (e.g., resulting from a partial dimension reduction in [2]) geometric conceptions fail to describe the coupling of the solver components and domains involved. Based on a generalized definition of inter-element connectivity we develop a new interface to the topological coupling of grids. We present its implementation in DUNE and give numerical results.

Bibliography

- [1] P. Bastian, G. Buse, O. Sander, Infrastructure for the Coupling of DUNE Grids, In: *Numerical Mathematics and Advanced Applications 2009*, Springer (2010), 107–114.
- [2] E. Miglio, S. Perotto, F. Saleri, Model Coupling Techniques for Free-surface Flow Problems: Part I, *Nonlinear Analysis: Theory, Methods & Applications* **63** (2005), e1885–e1896.
- [3] S. Müthing, P. Bastian, DUNE-MULTIDOMAINGRID: A Metagrid Approach to Subdomain Modeling, In: *Advances in DUNE*, Springer (2012), 59–73.
- [4] A. Quarteroni, A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications (1999).

^aUniversity of Freiburg, christoph.gersbacher@mathematik.uni-freiburg.de

2.25 Adaptive solution strategies for coupled porous media flow / free flow systems

Christoph Grüninger^a, Thomas Fetzer^a, Bernd Flemisch^a, Rainer Helmig^a

In order to solve many environmental and technical problems, flow in porous media must be coupled with free flow. We focus on the modeling and interpretation of evaporation from porous media under the influence of atmospheric processes, such as wind and radiation. We already have a two-dimensional REV-scale model concept which allows the monolithic coupling of a laminar single-phase free flow and a two-phase porous-medium flow under non-isothermal, compositional flow conditions [1, 2].

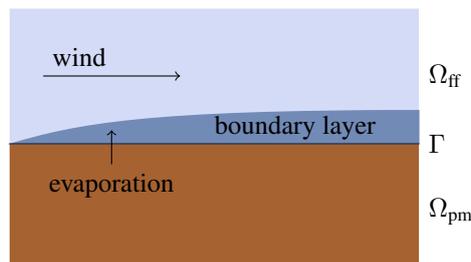


Figure 2.1: Setting and division of the domain in subdomains for free flow and porous medium flow with the coupling interface in between

We target field scale problems, but are only capable of simulating lab scale problems. Therefore a main goal of this work are tools for speedup. The physical processes that occur in both compartments are very different in space and time. In the free flow we expect turbulent behavior, which includes small scale effects with high velocities. In the porous medium, on the other hand, we have to model the much slower flow processes.

The monolithic concept of our coupling ties both compartments together, which leads to long simulation runs. In the course of this work, we look for strategies to gradually decouple the two different compartments in space and time. We want to couple the spacial discretization by a mortar approach in order to allow different resolved grids at the interface. For the approximation in time we intent an adaptive time discretization scheme such that the free flow can be simulated with smaller time steps on a finer grid. Overall it is our goal to increase the efficiency and the robustness of the simulation using a new, sequential coupling approach.

With the resulting software we will be able to simulate more realistic lab experiments. That will help to push the understanding of evaporation processes a bit further. With the gained robustness we hope to extend the laminar flow to turbulent flow and use a Reynolds averaging with an algebraic turbulence model or $k-\varepsilon$ model. This would open the door to field scale simulations.

Bibliography

- [1] K. Baber et al. *Numerical scheme for coupling two-phase compositional porous-media flow and one-phase compositional free flow*, IMA J. Appl. Math., **77**, 887–909, (2012).
- [2] K. Mosthaf et al. *A coupling concept for two-phase compositional porous-medium and single-phase compositional free flow*, Water Resour. Res., W10522, 19 PP, (2011).

^aInstitute for Modelling Hydraulic and Environmental Systems, University of Stuttgart, Germany, christoph.grueninger@iws.uni-stuttgart.de

2.26 FEM-FEM and FEM-BEM Coupling within the Dune Computational Software Environment

Alastair Radcliffe^a, Andreas Dedner^b

Some simple iterative solution schemes, and their implementation within the "Dune" computational development framework, will be presented for the coupling of two structured finite element meshes.

The exchange of differing combinations of Dirichlet and Neumann data between the finite element (fem) schemes built on each mesh will be studied.

Both overlapping and non-overlapping Schwartz type schemes will be discussed with an emphasis on their suitability for future finite element / boundary element (bem) coupling schemes, where calls to the "Bempp" boundary element software library will be made from within the Dune computational software environment for a wide range of applications in engineering and medicine.

Finally, a presentation of the convergence rates for some of the fem-fem coupling schemes applied to poisson type problems in 2-D and 3-D using simple structured meshes will be given, and their implications for the future work discussed.

^aUniversity of Warwick, UK. a.radcliffe@warwick.ac.uk

^bUniversity of Warwick, UK. a.s.dedner@warwick.ac.uk

2.27 Taking advantage seamlessly of new heterogeneous architectures, with a C++ DSEL to implement lowest-order methods for diffusive problem on general meshes

J-M. Gratien^a

Industrial simulation software have to manage: (i) the complexity of the underlying physical models, (ii) the complexity of numerical methods used to solve the PDE systems, and finally (iii) the complexity of the low level computer science services required to have efficient software on modern hardware. Nowadays, some frameworks like `Dune` [4, 6, 5] offer a number of advanced tools to deal with the complexity related to parallelism, meshes, linear solvers in a transparent way. However, high level complexity related to discretization methods and physical models lack of tools to help physicists to develop complex applications. Generative programming and domain-specific languages (DSL) are key technologies allowing to write code with a high level expressive language and take advantage of the efficiency of generated code for low level services. Their application to Scientific Computing has been up to now limited to Finite Element (FE) methods and Galerkin methods, for which a unified mathematical framework has been existing for a long time, (see projects like `Freefem++`, `Getdp`, `Getfem++`, `Sundance`, `Feel++` [3], `Fenics` project).

In reservoir and basin modeling, lowest order methods are promising methods allowing to handle general meshes. The unified mathematical frame for FV multi-points scheme and DFM/VFMH presented in [1, 2] allows us to extend the DSL used for FE and Galerkin methods to lowest order methods. We focus then on the capability of such language to allow the description and the resolution of various and complex problems with different lowest-order methods. We validate the design of the DSL that we have embedded in C++, on the implementation of several academic problems with various methods. We illustrate the benefits of such technology to handle seamlessly new heterogeneous architectures with multi-core processors enhanced by GPU-GPUs, thanks to the clear separation between the high level of abstractions enabling to design numerical methods and the low level layers required to have performance on new hybrid architectures.

Bibliography

- [1] Droniou, J. and Eymard, R. and Gallouët, T. and Herbin, R., A unified approach to mimetic finite difference, hybrid finite volume and mixed finite volume methods, *Mathematical Models and Methods in Applied Sciences*,20,2, pp.265-295,2010
- [2] Di Pietro, D. A. and Gratien, J.-M., Lowest order methods for diffusive problems on general meshes: A unified approach to definition and implementation, *Finite Volumes for Complex Applications VI*,Springer-Verlag,Fořt, J. and Fürst, J. and Halama, J. and Herbin, R. and Hubert, F., 2011, pp.3-19
- [3] Christophe Prud'homme, Life: Overview of a unified c++ implementation of the finite and spectral element methods in 1d, 2d and 3d, In *Workshop On State-Of-The-Art In Scientific And Parallel Computing*, Lecture Notes in Computer Science, page 10. Springer-Verlag, 2007
- [4] Dedner, A. and Klöforn, R. and Nolte, M. and Ohlberger, M., *DUNE-FEM* Web page
- [5] Markus Blatt and Peter Bastian, The Iterative Solver Template Library, *Applied Parallel Computing. State of the Art in Scientific Computing*, Bo Kågström and Erik Elmroth and Jack Dongarra and Jerzy Waśniewski,2007,vol 4699, series Lecture Notes in Computer Science

^aIFP Energies nouvelles,1 et 4 av Bois-Preau, 92500 Reuil-Malmaison, jean-marc.gratien@ifpen.fr

- [6] Bastian, P. and Blatt, M. and Dedner, A. and Engwer, C. and Klöfkorn, R. and Ohlberger, M. and Sander, O., A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part I: Abstract Framework, *Computing*, vol.82, pp.2-3, pp.103–119, 2008

2.28 Hybrid Parallelization of Assembly in Dune

Jorrit Fahlke^a, Christian Engwer^a

Upcoming exa-scale computers will exhibit multiple levels of concurrency: message passing between nodes (commonly MPI), multiple execution units within one node (thread-parallelism), and inherent parallelism provided by a single execution unit (SIMD). To gain a significant part of the peak performance, it is necessary to exploit all levels of parallelism.

In numerical code the main effort is the assembly and the solution of sparse linear systems. Linear algebra allows to hide specialized implementations behind a coarse grained interface. In contrast to this the assembly of vectors and matrices poses great challenges for general purpose PDE frameworks, since the code in the inner loops must be provided by users and cannot be optimized by the framework developers.

We use the Dune framework[1] and the Dune-PDELab discretization module. These already offer message-passing parallelism for different structured and unstructured grid managers, showing good performance. In order to support thread parallel and vectorized computations, the current interfaces of dune-grid and the assembling in Dune-PDELab must be extended. Partitioning of the grid between threads and data access have significant impact on the scalability for many-core systems. Vectorization is currently impeded by the design of the PDELab interfaces.

We present recent results to enable all levels of concurrency for general Dune applications and discuss the implications on the design of the grid and assembler interfaces.

Bibliography

- [1] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöforn, R. Kornhuber, M. Ohlberger, and O. Sander. A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part II: Implementation and Tests in DUNE. *Computing*, 82(2–3):121–138, 2008.

^aInstitute for Numerical and Applied Mathematics, University of Münster. {jorrit@jorrit.de, christian.engwer@uni-muenster.de}

2.29 Parallel Geometric Multigrid in deal.II

Timo Heister^a, Guido Kanschat^b

While deal.II (see [1]) supported massively parallel computations on locally refined meshes for several years and has been shown to scale to 16k cores [2], only preconditioners based on the system matrix (through PETSc and Trilinos) could be used. Here, we present recent development of parallel geometric multigrid in the deal.II library based on previous work for serial, locally refined meshes, see [3].

The method supports the host of different finite element spaces inside the library (for example continuous and discontinuous Galerkin) of arbitrary order. In this talk we describe how we balance the multigrid hierarchy and other parallel algorithms. We demonstrate the parallel scalability in numerical results.

Bibliography

- [1] W. Bangerth, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, T. D. Young. The deal.II Library, Version 8.1. *arXiv preprint <http://arxiv.org/abs/1312.2266v4>*, 2013.
- [2] W. Bangerth, C. Burstedde, T. Heister, M. Kronbichler. Algorithms and Data Structures for Massively Parallel Generic Finite Element Codes. *ACM Trans. Math. Softw.*, Volume 38(2), 2011.
- [3] B. Janssen, G. Kanschat. Adaptive Multilevel Methods with Local Smoothing for H^1 - and H^{curl} -Conforming High Order Finite Element Methods. *SIAM J. Sci. Comput.*, vol. 33/4, pp. 2095-2114, 2011.

^aClemson University, heister@clemson.edu

^bIWR, Universität Heidelberg, kanschat@uni-heidelberg.de

2.30 Preparing DUNE for future HPC architectures

Steffen Müthing^a, Peter Bastian^b, Dirk Ribbrock^c, Dominik Göddeke^d

Current and upcoming HPC architectures are characterized by a massive increase in the amount of parallelism. This trend is apparent at all scales, ranging from instruction level parallelism (SIMD) over increasingly large numbers of hardware threads (both in current mainstream CPUs and to an even larger extent in accelerators like GPGPUs and Intel MIC) up to the constantly growing number of nodes in current high-end computers.

We present recent enhancements to the DUNE framework and to PDELab that aim at exploiting this development across all those scales. In particular, we extend the existing parallel computation support in DUNE (based on classical MPI domain partitioning) to a hybrid model with shared memory parallelism on multi-threaded CPU nodes and accelerators like GPGPUs or Intel MIC. At the current stage, we mostly concentrate on the linear algebra portion of the framework. Moreover, we investigate the effects of leveraging the SIMD capabilities of current hardware at the different solver stages (assembly / linear algebra). Initial performance results are shown for a scalable porous media flow and transport solver with heterogeneous parameter fields and a Discontinuous Galerkin discretization, where we employ a highly efficient factorization approach with support for SIMD to improve the performance of the higher-order DG assembly and a set of hybridized, SIMD enabled linear algebra implementations.

^aIWR, Heidelberg University, steffen.muething@iwr.uni-heidelberg.de

^bIWR, Heidelberg University, peter.bastian@iwr.uni-heidelberg.de

^cFaculty of Mathematics, TU Dortmund, dirk.ribbrock@math.tu-dortmund.de

^dFaculty of Mathematics, TU Dortmund, dominik.goeddeke@math.tu-dortmund.de