

# A flexible software framework for multi-physics simulations with DUNE

Steffen Müthing  
Institut für Parallele und Verteilte Systeme  
University of Stuttgart

October 17, 2011

Many problems in the focus of current research involve the simultaneous simulation of multiple physical models that interact by means of a set of coupling conditions. There is a large number of projects working on the simplification of such PDE simulations by providing feature-rich integrated software frameworks, like deal.II [1], FEniCS [6] or DUNE [3, 4]. While most of these efforts support multi-field scenarios on a common spatial domain, multi-domain simulations pose their own unique set of implementation challenges, like subdomain tracking, DOF management, interface matching / extraction and communication of subproblem solutions to facilitate coupling between the individual problems. Existing approaches to these problems mostly center around the idea of combining existing single-physics implementations into a unified multi-physics simulation by providing a common integration layer that handles tasks like load balancing, geometry matching and inter-problem communication (cf. [8]). Examples of this approach include MpCCI [7] and SIERRA [5].

I will present a novel approach to the design of a multi-domain multi-physics framework based on the DUNE discretization toolkit PDELAB [2]. PDELAB can be used to implement a broad range of numerical PDE solution schemes; in order to achieve this generality, it provides precise mathematical definitions of all objects involved in the construction of such a scheme, along with corresponding programming interfaces. I will show how this framework can easily be extended to add support for several multi-domain settings, allowing for large numbers of arbitrarily shaped and possibly overlapping subdomains, variables which are only defined on subsets of the spatial domain, mortar spaces on subdomain interfaces and multiple equations which may also be defined on possibly overlapping subdomains. Due to the clear separation of concerns in the framework, it is possible to reuse many parts of the single-domain framework unchanged in a highly efficient fashion, e.g. time integrators (including multi-step methods) and I/O facilities.

The heavily templated C++ – programming style of PDELAB tends to require a substantial learning effort, and while PDELAB automates a great number of tasks, writing the program-specific program parts still requires a lot

of boiler-plate code, a task which tends to be tedious and error-prone. In the second part of this talk, I will shortly introduce an ongoing effort to extend the domain specific language UFL from the FEniCS project with a compiler backend that generates PDELAB code, which allows for the specification of the mathematical problem in a very concise and natural fashion. A second advantage of this approach is the ability to automatically manipulate the UFL expression, e.g. to automatically derive and implement the Jacobian of a given residual.

## References

- [1] W. Bangerth, R. Hartmann, and G. Kanschat. *deal.II Differential Equations Analysis Library, Technical Reference*. <http://www.dealii.org>.
- [2] P. Bastian, F. Heimann, and S. Marnach. Generic implementation of finite element methods in the Distributed and Unified Numerics Environment (DUNE). *Kybernetika*, 46(2):294–315, 2010.
- [3] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöforn, R. Kornhuber, Mario Ohlberger, and Oliver Sander. A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part I: Abstract Framework. *Computing*, 82(2-3):103–119, 2008.
- [4] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöforn, R. 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, 2008.
- [5] H. Edwards. Managing complexity in massively parallel, adaptive, multi-physics applications. *Engineering with Computers*, 22:135–155, 2006.
- [6] A. Logg. Automating the finite element method. *Arch. Comput. Methods Eng.*, 14(2):93–138, 2007.
- [7] MpCCI website. <http://www.mpicc.de>.
- [8] Eric de Sturler, Jay Hoeflinger, Laxmikant V. Kalé, and Milind Bhandarkar. A new approach to software integration frameworks for multi-physics simulation codes. In *Proceedings of the IFIP TC2/WG2.5 Working Conference on the Architecture of Scientific Software*, pages 87–104, Deventer, The Netherlands, The Netherlands, 2001. Kluwer, B.V.