Hardware-oriented Numerics for PDE

Dominik Göddeke

Department of Mathematics, TU Dortmund

Scientific computing is in the middle of a paradigm shift in the underlying hardware, and higher performance is only achieved by specialisation, heterogeneity and parallelisation. The number of cores on traditional architectures is increasing rapidly, and graphics processors (GPUs) can be considered forerunners of this trend, as they are designed to maximise throughput of many similar tasks rather than latency of individual tasks with a wide-SIMD design and massive on-chip fine-grained parallelism. In addition, the memory wall problem continues to increase, which is important not only for algorithms with poor arithmetic intensity.

We highlight that thinking in terms of sequential algorithms is a legacy we ultimately need to dispose of, as parallelism is abundant and the more natural approach. Traditional numerical approaches are often in contrast to this hardware evolution, as they have typically been designed for sequential execution, running faster and faster during the past years of frequency scaling. *Hardware-oriented numerics* is a research area in applied mathematics which aims to maximise numerical performance (fast h-independent convergence, robustness for a wide range of parameter settings, flexibility wrt. arbitrary configurations) without sacrificing the available peak performance: We claim that this goal can not be achieved by re-compiling or simply re-implementing existing numerical software, but that significant novel research is required.

After a detailed survey of hardware evolution, performance bottlenecks and associated programming models, we present several instructive examples highlighting various aspects of hardware-oriented numerics. We cover the analysis of existing numerical schemes, the design of novel ones, and important implementational details such as the choice of data structures and blocking techniques to increase locality and data reuse. We focus on the solution of large sparse linear systems arising from the discretisation of the underlying PDEs with FE/FD/FV-like techniques: Multigrid solvers are the only algorithmically scalable and asymptotically optimal scheme for such systems, and are thus a fundamental building block in the solution of PDE problems. Combined with appropriate finite element techniques, multigrid methods are very beneficial in terms of accuracy, scalability and performance.

The examples we present include mixed precision methods, patchwise general tensor product meshes, unstructured meshes, parallelisation strategies for inherently sequential and recursive numerically strong smoothers and preconditioners, and the important trade-off between global coupling and degree of available parallelism on all levels of heterogeneous systems.

References

[1] S. Turek *et al.*, 'FEAST – Realisation of hardware-oriented Numerics for HPC simulations with Finite Elements', Concurrency and Computation: Practice and Experiment **22(6)** (2010), 2247–2265.

[2] D. Göddeke and R. Strzodka, 'Mixed Precision GPU-Multigrid Solvers with Strong Smoothers', chap. 7 in Scientific Computing with Multicore and Accelerators, CRC Press (2010).

[3] M. Geveler *et al.*, 'Towards a complete FEM-based simulation toolkit on GPUs: Unstructured Grid Finite Element Geometric Multigrid solvers with strong smoothers based on Sparse Approximate Inverses', accepted in Computers and Fluids (2011).