
Miriam Mehl
Parallel Multigrid – a Scalable Alternative to Parallel CG?

Department of Computer Science
Technische Universität München
Boltzmannstr 3
85748 Garching
`mehl@in.tum.de`

From a computer science point of view, the performance evaluation of iterative solvers for large systems of equations does not only include numerical efficiency but also parallel scalability. Whereas multigrid methods are clearly superior from a pure numerical point of view, the majority of parallel solvers available in software packages are based on cg-like methods. These solvers achieve very good scalability on high performance computing platforms even for large processor numbers. In contrast, multigrid methods seem to perform much worse in the parallel context. This arises the question whether they are only a nice theoretical concept but not applicable in an efficient way in supercomputing practice.

Considerations that might lead to such a conclusion often only include so-called strong-scalability results — constant problem size and growing number of processors. This point of view does not take into account the advantage of multigrid methods — the constant number of iterations with increasing problem size. Thus, the correct setting must be to consider weak-scalability, that is problem sizes growing according to the processor number, and, in addition, time until convergence instead of only single solver iterations.

This approach leads to a new definition of an optimal solver. Whereas an optimal sequential solver computes the solution of a system in a computational time growing proportional to the number of unknowns, parallelisation even allows to look for solvers with constant computational time. So the question is: Is it possible to implement a solver that solves a system of equations in constant time independent from the size of the system or the mesh resolution, respectively, by suitably increasing the number of processors? In the optimal case, the number of processors increases only proportional to the problem size.

We will present first a theoretical comparison of cg-like and multigrid methods for parallel computations under simplifying assumptions showing the general potential of multigrid and cg-like solvers on supercomputers and also answering the question on the possibility to develop an optimal parallel solver. In addition, we will relate various parallel multigrid approaches from literature (see for example [1, 2, 3, 4, 6, 7]) to this theoretical analysis and, finally, present results from an own parallel multigrid implementation on adaptive Cartesian grids with tree-structure (see [5] for a description of the corresponding sequen-

tial code). In contrast to the purely theoretical analysis that only takes into account computational and communication costs, this will also give hints on ways to improve other performance aspects such as cache-efficiency that can be an additional stumbling block for high performance multigrid implementations due to the more complex data dependencies compared to cg-like solvers.

- [1] M. Adams and J. W. Demmel. Parallel multigrid solver for 3d unstructured finite element problems. In *Supercomputing '99: Proceedings of the 1999 ACM/IEEE conference on Supercomputing*, page 27, New York, NY, USA, 1999. ACM.
- [2] M. Berger, M. Aftosmis, and G. Adomavicius. Parallel multigrid on cartesian meshes with complex geometry. In *Proc. of the 8th Intl. Conf. on Parallel CFD. Trondheim*, 2000.
- [3] T. Gradl and U. R de. High performance multigrid on current large scale parallel computers. In *9th Workshop on Parallel Systems and Algorithms (PASA), Dresden, 26.02.2008*, volume 124 of *GI Edition: Lecture Notes in Informatics*, pages 37–45. Gesellschaft f r Informatik, 2008.
- [4] H. van Emden and U.Meier Yang. Boomeramg: A parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41(1):155–177, 2002.
- [5] M. Mehl, T. Weinzierl, and Ch. Zenger. A cache-oblivious selfadaptive full multigrid method. *Numerical Linear Algebra with Applications*, 13(2-3):275–291, 2006.
- [6] W. F. Mitchell. A refinement-tree based partitioning method for dynamic load balancing with adaptively refined grids. *Journal of Parallel and Distributed Computing*, 67(4):417–429, 2007.
- [7] G. W. Zumbusch. Data parallel iterators for hierarchical grid and tree algorithms. In W. E. Nagel, W. V. Walter, and W. Lehner, editors, EuroPar, volume 4128 of *Lecture Notes in Computer Science*, pages 625–634. Springer, 2006.