
Kenneth Esler
Fully Accelerating Classical AMG with GAMPACK
(GPU Algebraic Multigrid PACKage)

Stone Ridge Technology
2015 Emmorton Rd Suite 204
Bel Air
MD 21015
`kesler@stoneridgetechnology.com`
Vincent Natoli
Yongpeng Zhang
Aleksander Samardzic

Algebraic multigrid (AMG) has become the defacto standard for the solution and preconditioning of difficult linear systems arising from elliptic and parabolic PDEs in a broad range of domains. Its robust convergence rates, linear complexity in system size, and ability to handle unstructured and heterogeneous problems make it a nearly indispensable tool in fields such as reservoir simulation.

In the last few years, the underlying computational architectures on which high performance computing is performed have begun to undergo a dramatic shift. As CPU clock frequencies have plateaued due to power constraints, researchers have been forced to seek performance growth through increasing parallelism. Multicore processors have provided an evolutionary approach for which the methodologies based on coarse-grained parallelism that were developed for distributed memory computers have proved mostly adequate.

GPUs, however, require the simultaneous execution of thousands of threads and a careful orchestration of memory access to achieve good performance. This degree of multithreading, in turn, requires a more fine-grained approach to parallelism. The regular access patterns and naturally parallel smoothing methods of geometric multigrid has allowed this method to benefit relatively easily from the increased memory bandwidth and FLOP rate of GPUs. In contrast, the "branchy" execution and largely random memory access patterns of AMG algorithms have made acceleration with GPUs extremely challenging.

Initially, sparse matrix methods as a whole were largely dismissed by the HPC community as inappropriate for acceleration on GPUs. Soon, however, researchers proved this belief false by demonstrating significant speedups by applying GPUs to iterative solvers with trivial or no preconditioning. More complex preconditioners were gradually ported, including those based on incomplete factorizations. Early work on accelerating AMG focused solely on the solution stage, and again dismissed AMG setup as impossible to accelerate. Later work

demonstrated that by selecting appropriate algorithms, constructing an AMG hierarchy on GPU was possible, though the speedups achieved were more modest than those typical of simpler algorithms.

In this talk, we discuss our efforts to fully accelerate AMG algorithms on GPUs. By selecting appropriate algorithms and through careful implementation, we find that it is possible to achieve a good acceleration of classical AMG algorithms in both the setup and solve stages. In particular, we find that partitioning the unknowns with a PMIS approach and making use of distance-2 interpolators can provide a robust hierarchy with ample fine-grained parallelism. Combining these algorithms with an appropriate selection of smoothers and Krylov acceleration yields a complete solver package with good speedup relative to available CPU AMG solvers. We give examples, primarily drawn from reservoir simulation, and compare performance with open-source CPU solutions. Finally, we discuss our work to provide a solution which scales across multiple GPUs.