Preprint
UCRL-JC-136477

# Experiences with *BoomerAMG:* A Parallel Algebraic Multigrad Solver and Preconditioner for Large Linear Systems

*V. E. Henson and U. M. Yang*

This article was submitted to 16th International Association for Mathematics and Computers in Simulation World Congress 2000 on Scientific Computation, Applied Mathematics and Simulation, Lausanne, Switzerland, August 21-25, 2000

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

**February 22, 2000**

# Experiences with *BoomerAMG:* a parallel algebraic multigrid solver and preconditioner for large linear systems

Van Emden Henson*  Ulrike Meier Yang*

## 1 Introduction

Algebraic multigrid (AMG) is an attractive choice for solving large linear systems $A\mathbf{x} = \mathbf{b}$ on unstructured grids [6, 2]. While AMG is applicable as a solver for a variety of problems, its robustness may be enhanced by using it as a preconditioner for Krylov solvers, such as GMRES. The sheer size of modern problems, hundreds of millions or billions of unknowns, dictates the use of massively parallel computers. AMG consists of two phases: the setup phase, in which smaller and smaller linear systems are generated by means of linear transfer operators (interpolation and restriction); and the solve phase, which employs a smoothing operator, such as Gauss-Seidel or Jacobi relaxation. Most of these components can be parallelized in a straightforward fashion; however, the coarse-grid selection, in which the grid for a smaller linear system is created on which the error can be approximated, is highly sequential. It is important to develop parallel coarsening techniques.

We briefly present here the coarsening algorithms used in the parallel AMG code *BoomerAMG* and summarize some performance results for those algorithms. A detailed discussion of the algorithms and numerical results will be found in [3].

## 2 The Classical AMG Coarsening Algorithm

The coarse grid must be chosen to accurately represent $\mathbf{e}$, a *smooth error*. Coarse-grid selection consists of determining a small subset of the points that may be used to interpolate the value of other points. The basic concept is that if the matrix coefficient $a_{ik}$ is such that $e_k$ is particularly important in determining the value of $e_i$ in the $i$th equation of the residual, then the point $k$ is said to *influence* the point $i$.

Classical AMG [6] uses a two-pass approach to partition the grid $\Omega$ into fine-grid points $\mathcal{F}$ and coarse-grid points $\mathcal{C}$. In the *first pass*, a "measure" $\lambda_i$ is assigned to each point $i \in \Omega$. Initially, $\lambda_i$ is the number of adjacent points influenced by the point $i$. A point $j$, having the global maximal $\lambda_j$, is declared to be a $\mathcal{C}$-point, and all points $k$ influenced by $j$ are declared to be $\mathcal{F}$-points. For each such point $k$, we increment the measure of any unselected neighbors influencing $k$. This selection process is repeated until all points have been assigned to either $\mathcal{C}$ or $\mathcal{F}$. A *second pass* is made through the points in $\mathcal{F}$. For each $j \in \mathcal{F}$, if another point $k \in \mathcal{F}$ is found such that either $j$ or $k$ influences the other, then one of the points is converted into a $\mathcal{C}$-point.

This algorithm suffers from the fact that it is inherently sequential. We describe several parallel coarsening schemes in the next section.

## 3 The Parallel Coarsening Schemes

The *RS scheme* is a parallel implementation of the classical coarsening algorithm. The classical scheme is run, independently, on each processor. The quality of the coarse grid generally suffers because no account has been taken of the relationships between points with influence across processor boundaries. Several different options can be considered. One possibility, of course, is to do nothing about the boundaries (this is the RS0 variant). Another possibility, for example, is that an extra "second" pass could be run only on points adjacent to the processor boundaries (denoted as the RS3 variant).

One drawback to the RS scheme is that it produces different coarsenings when the same problem is run on different numbers of processors. A second method, the *CLJP scheme* [1], which is based on modifications of parallel independent set algorithms [5, 4], produces the same coarsening for any number of processors. We begin by forming the "influence graph" of the matrix, whose vertices are the gridpoints, and which has a directed edge $i \to j$ if the point $i$ is influenced by the point $j$. To each point $j$ a measure is assigned, equal to the number of points influenced by $j$. Then a random number between 0 and 1 is added to each measure. The graph is scanned, and each vertex $i$ whose measure exceeds that of all points adjacent to $i$ is moved from $\Omega$ to the set $\mathcal{C}$. This can be done entirely in parallel, with each processor producing a set of $\mathcal{C}$-points having the required property. Since the variables at points in $\mathcal{C}$ will not be interpolated, we can eliminate the edges of the graph associated with points that influence the $\mathcal{C}$-points, and decrement the corresponding measures. Similarly, points in $\mathcal{C}$, having already been selected, have their measures set to zero, and the corresponding edges removed from the graph. If two points $i$ and $j$, not in $\mathcal{C}$, are both influenced by a common $\mathcal{C}$-point, and $i$ influences $j$, we remove the edge corresponding to that influence and decrement the measure for $i$. When the measure of any points not in $\mathcal{C}$ falls below unity, that point is assigned to $\mathcal{F}$. At this stage a communication step is employed, in which the new measures are updated and any edge removal pertaining to the boundary points is performed. The entire process is then repeated on the modified graph $G$, and this is continued until all points are in either $\mathcal{C}$ or $\mathcal{F}$.

Another drawback of the RS scheme is that it produces a coarsest-grid with a small number of points on each processor. When this occurs the coarsening stagnates, that is, further calls to the coarsening routine do not produce smaller coarse grids. As a result, the global coarsest-grid can be quite large even though the local coarsest-grids are quite small. A simple solution, called the *RS-hybrid scheme*, is to utilize the RS coarsening until it stagnates, then to switch to the CLJP algorithm and continue until a small global coarsest-grid has been achieved.

The last coarsening scheme we discuss, known as the *Falgout scheme*, is a combination of the RS and CLJP schemes. The basic idea is to use the RS approach, that is, to employ on each processor the classical AMG scheme, as a means of selecting the first "independent set" in the CLJP approach. After that first set is chosen, the CLJP heuristics are applied to the edges and measures, and the resulting graph is fed into the CLJP algorithm to complete the coarsening.

## 4 Numerical Results

Each of these coarsening schemes is incorporated into *BoomerAMG*, a parallel AMG code designed for use on massively parallel, distributed memory machines. The communication and parallelism are handled by use of MPI.

Tests are reported for problems involving the five-point (2D finite difference), nine-point (2D finite element), seven-point (3D finite difference), and twenty-seven point (3D finite element) Laplacian operators. All of these tests are performed on structured, Cartesian grids (unstructured-grid results will be reported in a future paper). The problem sizes are generally 64,000 points (40x40x40) per processor for 3-dimensional problems and 122,500 points (350x350) per processor for 2-dimensional problems, which is held constant as the number of processors increases. Results are available for up to 1024 processors.

In most cases, the Falgout and RS-hybrid coarsenings (RS0 and RS3) produce significantly better convergence rates and timings than the CLJP coarsening. In the case of the twenty-seven point Laplacian operator, the convergence rates of all the methods were fairly similar. For this particular case CLJP also achieves fairly low operator complexities, hence, the timings were also similar. In all other cases, however, CLJP coarsening produces noticeably larger operator complexities, making it the least desirable method.

For certain cases the RS0 coarsening is favored due to the fact that the test problems are structured; for example, the RS0 coarsening "fits" the seven-point Laplacian (with 40x40x40 points on each processor), yielding a convergence factor of 0.25. However, changing the local grid size to 39x39x39 points dramatically degrades the convergence rate to 0.5. The fact that RS0 ignores the boundaries eliminates any communication during coarsening, which in turn produces excellent setup times. No approach that treats the boundaries can compete with the setup times of RS0, assuming it generates coarse grids of comparable sizes. Consequently, whenever a good convergence results from the RS0 coarsening, this is the method of choice.

## 5 Conclusions

The RS-hybrid, CLJP, and Falgout coarsening schemes can all be used to coarsen effectively in a parallel AMG code. Overall, the Falgout coarsening appears to be the most robust and best overall coarsening strategy for structured-grid test problems. Further improvements are likely to develop, especially as the research moves into the realm of unstructured-grid problems.


## Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract number W-7405-Eng-48.


## References


[1] A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, *Coarse-grid selection for parallel algebraic multigrid*, in Proceedings of the Fifth International Symposium on Solving Irregularly Structured Problems in Parallel, Lecture Notes in Computer Science vol. 1457, Springer-Verlag, New York, 1998, pp. 104-115.

[2] A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteuffel, S.F. McCormick, G.N. Miranda, J.W. Ruge, *Robustness and scalability of algebraic multigrid*, SIAM J. Sci. Stat. Comput., to appear.

[3] V.E. Henson, U.M. Yang, *BoomerAMG: a parallel algebraic multigrid solver and preconditioner for large linear systems*, in preparation, to appear in a Special Issue of Applied Numerical Mathematics.

[4] M.T. Jones, P.E. Plassman, *A parallel graph coloring heuristic*, SIAM J. Sc. Stat. Computing, 14 (1993), pp. 654-669.

[5] M. Luby, *A simple parallel algorithm for the maximal independent set problem*, SIAM J. on Computing, 15 (1986), pp. 1036-1053.

[6] J.W. Ruge, K. Stüben, *Algebraic multigrid (AMG)*, in Multigrid Methods (ed. S.F. McCormick), Frontiers in Applied Mathematics vol. 3, SIAM, Philadelphia, PA, 1987, pp. 73-130.