Discretization-Accuracy Convergence for Full Algebraic Multigrid

Wayne Mitchell

Abstract

Full multigrid (FMG) is well known for converging to the level of discretization accuracy in a single cycle on a wide class of partial differential equations when the multigrid hierarchy is derived from problem geometry. When applying an FMG cycle to a hierarchy generated by algebraic multigrid (AMG), however, this scalable convergence to discretization accuracy may be lost. This paper examines the cause of this phenomenon and explores some improvements to standard AMG interpolation which can restore single cycle convergence to discretization accuracy.

1 Introduction

For simple problem geometries and structured meshes, geometric multigrid (GMG) algorithms are straightforward to implement and yield scalable solvers and preconditioners for a wide class of partial differential equations. In particular, full multigrid (FMG) is well known for converging to discretization accuracy in a single cycle with $O(n)$ computational cost. When meshes become more complicated, however, geometric methods can become prohibitively difficult, and a need arises for methods which rely only on the discrete operator for a problem and not the problem geometry. Algebraic multigrid (AMG) fulfills this requirement, generating a multigrid hierarchy from only the fine grid operator and delivering V-cycle convergence factors comparable to those produced by geometric multigrid. Applying the full multigrid algorithm to a hierarchy generated by AMG (denoted FAMG) does not generally obtain discretization accuracy in a single cycle, however. This means that as problem size grows, more V-cycles (and consequently more computational effort) are required in order for AMG to solve the problem as opposed to the fixed, single-cycle cost of FMG. In addition, more complex parallel algorithms (such as domain and range decomposition algorithms [1]) which are built on top of AMG hierarchies have little hope of scalable convergence to discretization accuracy if the underlying AMG hierarchies do not perform well in the simpler FAMG setting. The following section examines why FAMG fails to achieve the same kind of convergence to discretization accuracy as FMG. Section 3 then presents two approaches for improving the AMG hierarchy in order to restore this convergence.

2 AMG Interpolation Error

Throughout this paper, "standard" AMG refers to the Ruge-Stüben method for generating an AMG hierarchy [2, 5]. Coarse grids are selected by a coloring algorithm and the coarse grid operators are formed via $A^{2h} = P^T A^h P$, where A^h is a fine grid operator and P is the interpolation operator between coarse and fine grids. The entries of P itself are interpolation weights defined through the following formula:

$$w_{ij} = \frac{a_{ij} + \sum\limits_{k \in F_i^s} a_{ik} \frac{a_{kj}}{\sum\limits_{j' \in C_i} a_{kj'}}}{a_{ii} + \sum\limits_{m \in F_i^w} a_{im}}, \tag{1}$$

where w_{ij} is the interpolation weight from a coarse node j to fine node i, the a's are entries from the matrix A^h on the fine level, C_i are the coarse points strongly connected to fine node i, F_i^s are the strongly connected fine points, and F_i^w the weakly connected points. Also, all FMG or FAMG cycles described below use a single V-cycle on each level during the nested iteration process unless otherwise noted.

The following analysis of standard AMG considers a Poisson equation discretized on a square with bilinear finite elements as a model problem:

$$\begin{aligned} -\Delta u &= f \,,\; u \in \Omega = [-1,1] \times [-1,1] \\ u &= 0 \,,\; u \in \partial\Omega. \end{aligned}$$

A manufactured solution, $u(x,y) = (x+1)(1-x)(y+1)(1-y)$, yields right hand side $f(x,y) = 2((y+1)(1-y)+(x+1)(1-x))$. For such a problem, standard AMG is known to have excellent V-cycle convergence, but standard FAMG fails to converge to discretization accuracy in a single cycle. Figure 1 shows convergence of the relative total error,

$$err = \frac{||u_i^h - u||}{||u_{initial}^h||},$$

for FMG vs. FAMG, where u is the true solution evaluated on the fine grid, u_i^h is the solution obtained by the nested iteration process plus i additional V-cycles, and $u_{initial}^h$ is a random initial guess. This discussion uses the l^2 norm throughout. Note that the error shown stalls at the level of discretization error since it is measured against an analytic solution u. As the grid is refined (i.e. as problem size increases), FAMG yields less accurate fine grid solutions after the nested iteration process, thus requiring more fine grid V-cycles to obtain the level of discretization accuracy.

Note that for the model problem with the appropriate coloring scheme, the coarse grids generated in the AMG hierarchy are the same as the coarse grids generated by GMG. The

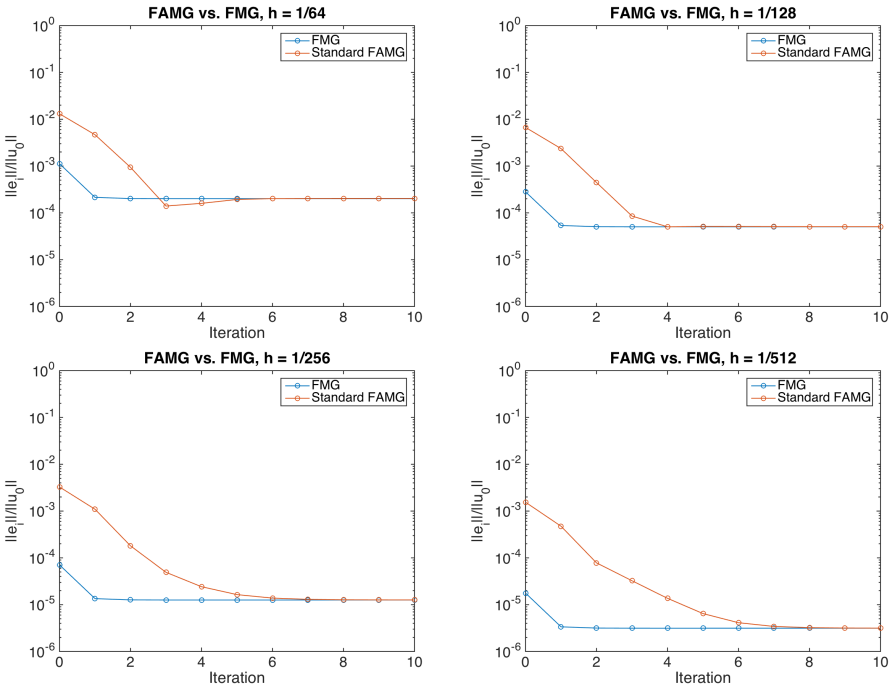

Figure 1: Relative total error convergence for standard FAMG vs. FMG.

interpolation operators between these grids are also very similar between AMG and GMG, in fact these operators only differ in the way they choose interpolation weights near the boundary of the domain. In order to obtain FMG convergence to discretization accuracy, however, there is a very specific requirement on the interpolation error between grids: the interpolation error, $||u^h - Pu^{2h}|| = ||A^{-1}f - P(P^TAP)^{-1}P^Tf||$, on each level of the multigrid hierarchy must scale with the same order of accuracy as the discretization. So for the model problem, the l^2 interpolation error should scale like $O(h)$. Figure 2 shows the interpolation error on each level (with 0 representing the finest level) for FMG and FAMG. These errors are calculated with the manufactured right hand side f. FMG obtains exactly $O(h)$ convergence of the interpolation error, while FAMG fails this requirement. Though the difference in interpolation operators between the two methods is subtle (again, they differ only in the way they choose weights at the boundary), they achieve very different behavior across the levels of the hierarchy.

A visualization of the error, $u^h - Pu^{2h}$, over the domain is helpful in better understanding the nature of interpolation error for FMG vs. FAMG. As seen in Figure 3, the interpolation error for FMG is small and oscillatory. This is precisely the sort of error that is effectively removed by a V-cycle on the fine grid. For FAMG, however, there is a large smooth component to the error produced by large discrepancies near the boundary which

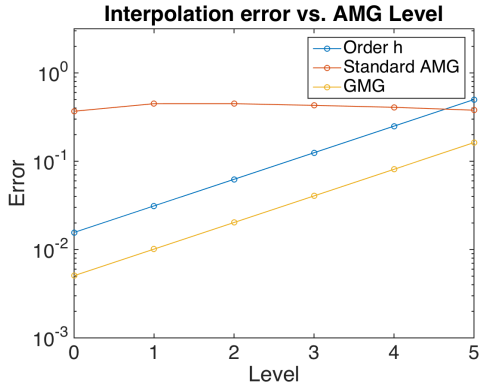

Figure 2: Interpolation error across levels of the multigrid hierarchies generated by standard AMG vs. GMG (with 0 representing the finest level).

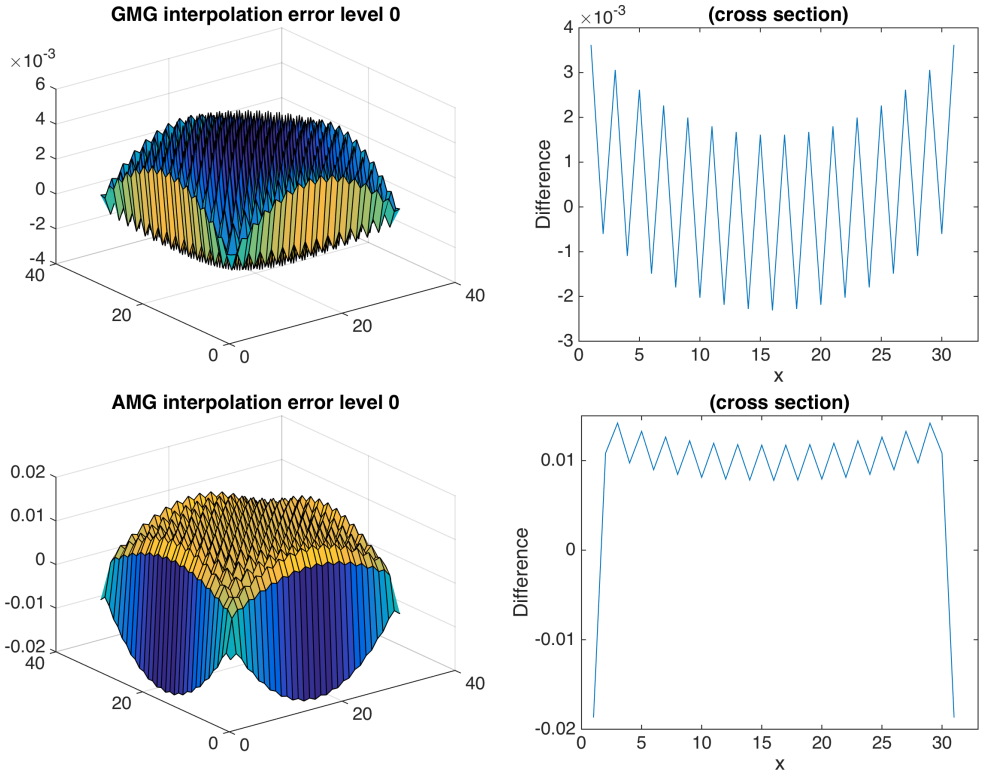

Figure 3: Plot of the finest level interpolation error over the grid for GMG (top) vs. standard AMG (bottom). The cross section is taken across the middle of the domain.

pollute the entire domain. More V-cycles are required to remove this smooth mode in the error.

The above analysis all suggests that FAMG might be improved by changing interpolation in some way to recover $O(h)$ scaling of the interpolation error. The following section describes two changes to interpolation which accomplish this goal for at least the model problem.

3 Improving AMG Interpolation

In order to improve the properties of interpolation for AMG, the goal is to ensure the appropriate vectors lie in the range of interpolation on each level. The standard Ruge-Stüben way of choosing interpolation weights in AMG ensures that the constant vector is in the range of interpolation. The analysis of the previous section shows that this property is insufficient for achieving good interpolation near the boundary. Thus the range of interpolation must be enriched in some way to recover good interpolation everywhere in the domain. The following subsections approach this problem in two different ways.

3.1 Fitting smooth vectors

One approach to improving the range of interpolation is to consider how well interpolation fits near-null space vectors for the operators on each level. As mentioned above, the Ruge-Stüben formula for interpolation weights fits the constant vector. Interpolation may fit an arbitrary vector $\mathbf{x}$ by using the following formula from [3] for the interpolation weights instead of equation (1):

$$w_{ij} = \frac{a_{ij} + \sum\limits_{k \in F_i^s} a_{ik} \frac{a_{kj} x_k}{\sum\limits_{j' \in C_i} a_{kj'} x_{j'}}}{a_{ii} + \sum\limits_{m \in F_i^w} a_{im}}. \tag{2}$$

Note that letting $\mathbf{x} = \mathbf{1}$ yields the Ruge-Stüben formula for interpolation weights.

For the model Poisson problem, the eigenvectors are known. Thus let $\mathbf{x}$ be the eigenvector associated with the smallest eigenvalue (the sine hump). As shown in Figure 4, defining interpolation in this way restores the order h convergence of the interpolation error across the multigrid levels and yields FAMG convergence to discretization accuracy in a single cycle. In practice, however, the near-null space of any given operator is generally unknown a priori. A fully adaptive method could be used to find the near-null space, but such methods are expensive.

A cheaper approach is to use the idea of trying to fit the local null space at each node. For the model problem, the operator A^h has row sum zero for all interior nodes, so fitting a constant vector fits the local null space at these nodes. Near the boundary

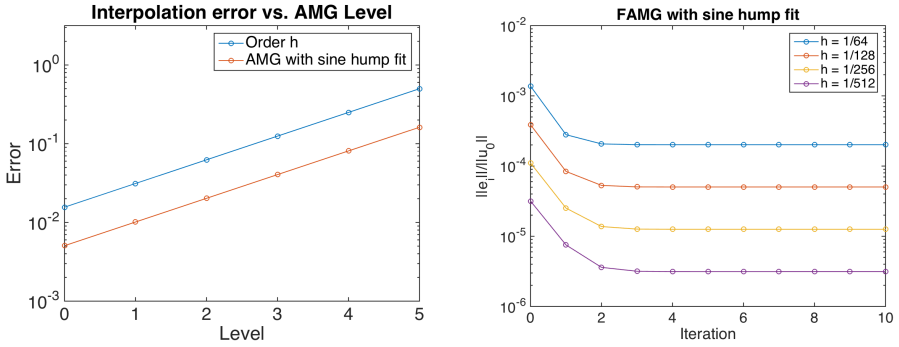

Figure 4: Interpolation error across multigrid levels and FAMG convergence when choosing interpolation to fit the sine hump.

of the domain, however, the operator does not have row sum zero, and the constant does a poor job of approximating the local null space. A better approach is to fit a constant vector which has been smoothed. Thus interpolation is constructed on each level using (2) as the formula for the weights and choosing $\mathbf{x}$ to be a constant vector wherever the row sum of the operator is zero and a smoothed constant vector otherwise. This has the effect of modifying interpolation only near the boundary (where interpolation needs to be improved). As shown in Figure 5, this method also restores order h convergence of the interpolation error and convergence to discretization accuracy in a single cycle for FAMG. As the problem size increases, however, the smoothed constant vector requires more smoothing iterations in order for the method to perform well. When $h = 1/64$, one Jacobi iteration was sufficient, whereas for $h = 1/512$, eight Jacobi iterations were required to obtain the results shown.

It is also unclear how robust this method of choosing interpolation will be across different problems and domain geometries. When choosing a different coarsening scheme which does not yield the same grids as geometric coarsening for the model problem, this method of choosing interpolation fails to obtain discretization accuracy in a single FAMG cycle. This implies that while this very simple and cheap fix is sufficient for a very simple model problem with nice geometry, it will not be sufficient for more complicated problems.

3.2 Fitting linears

A more direct approach to improving the range of interpolation is to use some geometric information about the problem to fit piecewise linear vectors. Specifically, the interpolation weights will be based on the locations of the nodes. This means that the method will no longer be purely algebraic (i.e. it requires additional information besides just the fine grid matrix), but asking for the locations of nodes is not unreasonable in practice. This method will retain purely algebraic coarsening methods, as this is the main difficulty in

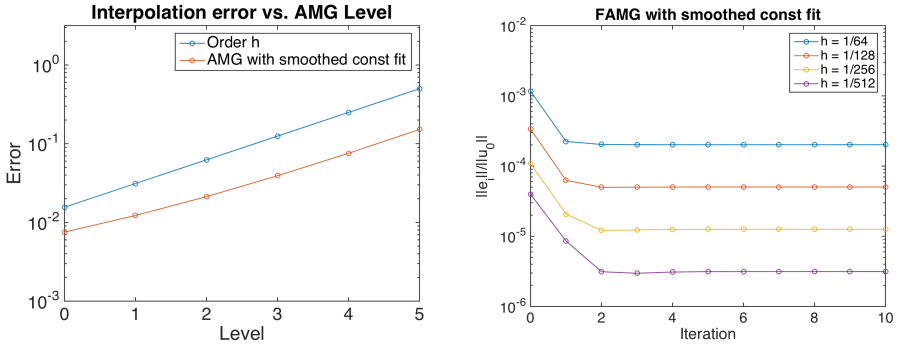

Figure 5: Interpolation error across multigrid levels and FAMG convergence when choosing interpolation to fit a constant vector where the operator on that level has row sum zero and a smoothed constant vector otherwise.

implementing geometric multigrid methods.

To construct interpolation for a given F-point, the strongly connected C-points are counted (this comprises the interpolary set), and the distance from the F-point to each C-point is measured. If the F-point is close to a boundary (i.e. the distance to the boundary is less than or equal to the maximum distance to another C-point), the boundary is added as a C-point to the interpolary set. Note that there is no need to actually calculate a weight for the point on the boundary, since the zero Dirichlet boundaries are eliminated from the matrix equations, but the number of C-points in the interpolary set will affect how other weights are calculated. Now the weights are chosen to exactly interpolate the constant vector, a linear vector in x, and a linear vector in y. For three or fewer C-points in the interpolary set, this is done by doing a least squares fit. For more than three C-points, standard AMG weights are calculated and then perturbed to fit the constant and linear vectors. The perturbations are calculated such that they are minimal changes to the original AMG weights while still exactly interpolating the constant and linear vectors.

As shown in Figure 6, using this method for interpolation yields discretization accuracy in a single FAMG cycle for the model problem. This method also shows promise for being more robust over more complex geometries and coarsening schemes. Figure 7 shows results for this method when using a coarsening scheme which yields different coarse grids from what would be obtained through geometric coarsening. The interpolation error across multigrid levels does not follow $O(h)$ convergence, but it does decrease nicely on the finest few levels. As such, the FAMG cycle required two V-cycles on each level in order to obtain the discretization-accuracy convergence shown (a single V-cycle on each level was not sufficient). This implies that, while constructing interpolation correctly is essential for FAMG convergence, choosing appropriate coarse grids and strength of connection matrices on each level is also an important factor in obtaining discretization accuracy.

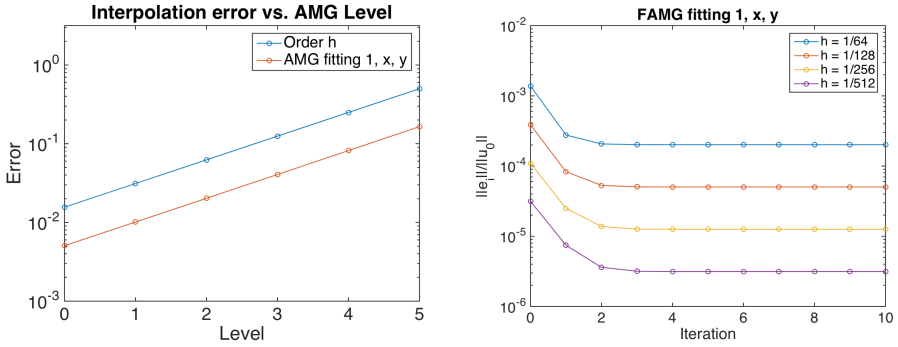

Figure 6: Interpolation error across multigrid levels and FAMG convergence when choosing interpolation to fit constant, linear in x, and linear in y vectors.

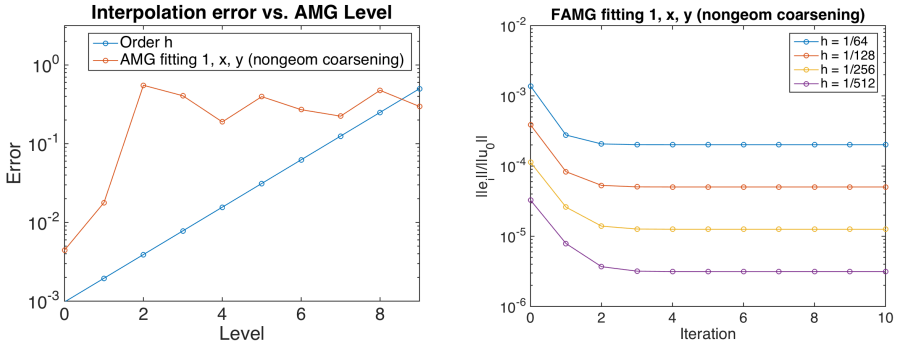

Figure 7: Interpolation error across multigrid levels and FAMG convergence when choosing interpolation to fit constant, linear in x, and linear in y vectors and using a coarsening scheme which does not yield geometric coarse grids. Note that the FAMG cycle used here employs two V-cycles on each level.

4 Conclusions

Applying a full multigrid cycle to a standard AMG hierarchy does not yield a solution at the level of discretization accuracy in a single cycle. Analysis of a model Poisson problem shows that the interpolation operators produced by AMG are insufficient to achieve the desired discretization-accuracy convergence. There are some different approaches to improving AMG interpolation in order to recover convergence to discretization accuracy. Fully adaptive methods which place the near-null space of the operators in the range of interpolation on each level should be successful, but these methods are expensive. Locally fitting a smoothed constant vector where the operator does not have row sum zero and a constant vector otherwise is a much cheaper and easier fix for the model problem with a standard coarsening scheme but is not a sufficiently robust method. Using geometric

information about the problem (namely the location of the nodes and the boundary of the domain) to explicitly attempt to fit linear vectors when constructing interpolation shows the most promise as a robust method for recovering convergence of FAMG to discretization accuracy in a single cycle. Constructing interpolation in this way produces a method that is no longer purely algebraic, but the required geometric information is readily available in most finite element codes. Note that this paper has not generally discussed coarsening schemes. Choosing appropriate coarse grids will also be an important factor in obtaining discretization accuracy on problems with more complicated geometries, so this topic deserves further examination.

References

[1] Bank R, Falgout R, Jones T, Manteuffel T A, McCormick S F, Ruge J W. Algebraic Multigrid Domain and Range Decomposition (AMG-DD/AMG-RD). *SIAM Journal of Scientific Computing*. 2015; 37(5), S113-S136.

[2] Brandt A, McCormick S F, Ruge J. Algebraic Multigrid (AMG) for sparse matrix equations. *Sparsity and Its Applications*. 1984.

[3] Brezina M, Falgout R, MacLachlan S, Manteuffel T, McCormick S, Ruge J. Adaptive Algebraic Multigrid. *SIAM Journal on Scientific Computing*. 2005; 27(4), 1261-1286.

[4] Briggs W L, Henson V E, McCormick S F. A Multigrid Tutorial. *SIAM*. 2000.

[5] Ruge J, Stüben K. Algebraic Multigrid (AMG). *Frontiers in Applied Mathematics*. 1987; Vol. 3 *Multigrid Methods*, 73-130.