LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-696160

Discretization-Accuracy Convergence for Full Algebraic Multigrid

R. D. Falgout, T. A. Manteuffel, S. F. McCormick, W. B. Mitchell

June 29, 2016

SIAM Journal on Scientific Computing

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

DISCRETIZATION-ACCURACY CONVERGENCE FOR FULL ALGEBRAIC MULTIGRID*

ROBERT D. FALGOUT†, THOMAS A. MANTEUFFEL‡, STEPHEN F. MCCORMICK‡, AND WAYNE B. MITCHELL‡

Abstract. Full multigrid (FMG) is well known for converging to the level of discretization accuracy in a single cycle for 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 is usually lost. This paper examines the cause of this loss and explores some improvements to standard AMG interpolation that can restore single-cycle convergence to discretization accuracy.

Key words. algebraic multigrid, full multigrid, discretization-accuracy solvers

1. Introduction. Full multigrid (FMG) is a solver for elliptic partial differential equations (PDE's) that is well known for converging to discretization accuracy in a single cycle with $O(n)$ computational cost, where n is the number of degrees of freedom [3, 10]. As such, FMG provides an optimal solver for elliptic PDE's. Though FMG is sometimes avoided in a parallel computing setting due to significant communication cost, it is actually competitive with multigrid V-cycles in parallel when a solution with accuracy on the order of the discretization (as opposed to some fixed tolerance) is desired. Recent work to ameliorate communication cost for FMG further increases its appeal as a solver for large elliptic PDE problems. Notable examples of this effort include the segmental refinement approach described in [1] as well as previous work optimizing the number of multigrid cycles performed on each level of an FMG cycle to ensure discretization accuracy is obtained with the minimal amount of effort [7, 9].

Previous study of FMG typically applies the algorithm in a geomegric multigrid (GMG) setting. For simple problem geometries and structured meshes, GMG algorithms can be straightforward to implement, but when meshes and other problem characteristics become more complicated, geometric methods can become prohibitively difficult, and a need arises for methods that rely only on the discrete operator for a problem and not the problem geometry. Algebraic multigrid (AMG) [4, 8] 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) is not well studied, however.

This paper reveals that FAMG does not generally obtain discretization accuracy in a single cycle. Even on simple problems for which AMG has excellent V-cycle convergence, applying an FAMG cycle can yield poor results: as problem size grows, increasingly more V-cycles (and consequently more computational effort) are usually required for AMG to solve the problem, with a cost that grows well beyond that of optimal, single-cycle FMG. This limits the capabilities of AMG as a discretization-accuracy solver. Recovering FMG-like convergence for FAMG would constitute a major advancement for algebraic methods, making them competitive with their geometric counterparts as discretization-accuracy solvers. In addition, more complex

*Submitted to the editors June 29, 2016. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract no. W-7405-Eng-48. Information management release number LLNL-JRNL-696160.

†Center for Applied Scientific Computing, Lawrence Livermore National Lab

‡Department of Applied Math, University of Colorado, Boulder

algorithms (such as domain and range decomposition algorithms [2]), 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. Thus, improving AMG hierarchies so that they perform better in a discretization accuracy sense is a crucial step towards obtaining scalable algebraic solvers for partial differential equations.

This paper aims to explain the difference in convergence characteristics of FMG and FAMG and to explore improvements to the standard AMG hierarchy that result in recovering convergence to discretization accuracy for FAMG. Section 2 begins by describing FMG and the assumptions under which it can be shown that FMG achieves convergence to discretization accuracy in a single cycle. Section 3 recasts the key interpolation assumption used in Section 2 in an algebraic framework by relating it to the strong approximation property. Section 4 then examines why FAMG fails to replicate the convergence displayed by FMG by studying a model Poisson problem. Section 5 provides details on how standard AMG constructs interpolation, and Section 6 then presents some approaches for improving AMG interpolation in order to achieve performance similar to FMG. Finally, section 7 discusses computational performance benefits of the improved FAMG cycle over standard AMG V-cycles with timing results verifying a reduction in computational cost from $O(n\log(n))$ to $O(n)$.

2. FMG Convergence. An FMG cycle, illustrated in Figure 1, solves a given problem by first solving its approximation on the coarsest grid, interpolating the solution up to the next finer grid, and then repeating this process (solving on each level by performing V-cycles) until a desired finest grid is reached. The number of V-cycles used to solve on each grid level should be constant (typically one or two), resulting in an overall $O(n)$ cost for the cycle. An inductive proof shows that such a cycle is capable of achieving discretization accuracy under certain assumptions on the accuracy of interpolation between grids [6].

To see the essence of this proof consider a pair of grids in the multigrid hierarchy with prolongation operator, P, interpolating from the coarse to fine grid. Let A^h and $A^{2h} = P^T A^h P$ be symmetric positive definite fine- and coarse-grid operators, respectively, and let $\mathbf{u}^h$ and $\mathbf{u}^{2h}$ be fine- and coarse-grid discrete solutions to the given problem:

$$A^h\mathbf{u}^h = \mathbf{f}^h$$
$$A^{2h}\mathbf{u}^{2h} = \mathbf{f}^{2h}.$$

The induction hypothesis states that the approximation of the solution on the coarse grid, $\mathbf{v}^{2h}$, has accuracy on the order of the grid $2h$ discretization:

$$||\mathbf{u}^{2h} - \mathbf{v}^{2h}||_{A^{2h}} \leq K(2h)^p,$$

where p is the discretization order. The A^{2h}-norm used here is the energy norm, defined as $||\mathbf{w}||_{A^{2h}} = \sqrt{\langle A^{2h}\mathbf{w}, \mathbf{w}\rangle}$, where $\langle\cdot,\cdot\rangle$ is the discrete L^2 inner product. A similar definition follows for the A^h-norm. This is simply for ease of presentation, since the argument may be made in any norm for which the assumptions presented here hold. Now assume that interpolation has accuracy on the order of the fine-grid discretization:

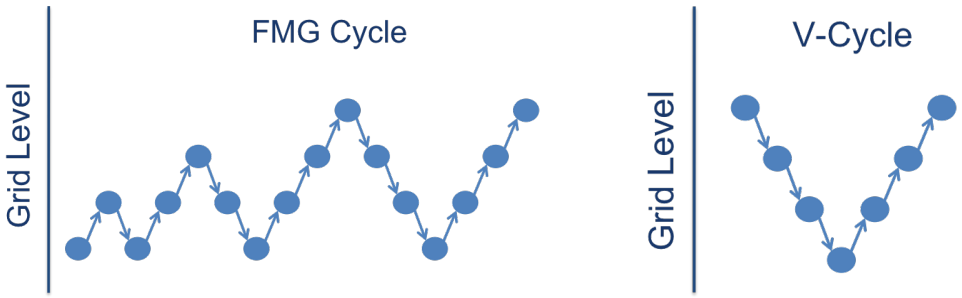

FIG. 1. *Illustration of cycle structures for FMG and a V-cycle.*

$$||\mathbf{u}^h - P\mathbf{u}^{2h}||_{A^h} \leq K\alpha h^p, \tag{1}$$

where P is the interpolation operator from grid $2h$ to grid h and α is a positive constant of $O(1)$. Putting these assumptions together yields a bound on the error present in the interpolated approximation $P\mathbf{v}^{2h}$:

$$\begin{aligned} ||\mathbf{u}^h - P\mathbf{v}^{2h}||_{A^h} &\leq ||\mathbf{u}^h - P\mathbf{u}^{2h}||_{A^h} + ||P\mathbf{u}^{2h} - P\mathbf{v}^{2h}||_{A^h} \\ &= ||\mathbf{u}^h - P\mathbf{u}^{2h}||_{A^h} + ||\mathbf{u}^{2h} - \mathbf{v}^{2h}||_{A^{2h}} \\ &\leq K\alpha h^p + K(2h)^p \\ &= K(\alpha + 2^p)h^p. \end{aligned}$$

Note that the equality in the second line above follows from the Galerkin condition ($A^{2h} = P^T A^h P$):

$$\begin{aligned} ||P\mathbf{u}^{2h} - P\mathbf{v}^{2h}||_{A^h} &= \langle A^h P(\mathbf{u}^{2h} - \mathbf{v}^{2h}), P(\mathbf{u}^{2h} - \mathbf{v}^{2h})\rangle \\ &= \langle P^T A^h P(\mathbf{u}^{2h} - \mathbf{v}^{2h}), (\mathbf{u}^{2h} - \mathbf{v}^{2h})\rangle \\ &= \langle A^{2h}(\mathbf{u}^{2h} - \mathbf{v}^{2h}), (\mathbf{u}^{2h} - \mathbf{v}^{2h})\rangle \\ &= ||\mathbf{u}^{2h} - \mathbf{v}^{2h}||_{A^{2h}}. \end{aligned}$$

Thus, given a V-cycle with convergence factor ρ, performing $\log_\rho(\alpha + 2^p) = O(1)$ cycles is sufficient to reduce the fine-grid error to the level of discretization accuracy: $||\mathbf{u}^h - \mathbf{v}^h||_{A_h} \leq Kh^p$.

3. Strong Approximation Property. The interpolation assumption (1) relies on the step size, h, and is therefore inherently tied to problem geometry. Under certain assumptions, however, it can be shown that (1) is implied by the purely algebraic bound known as the strong approximation property (SAP)[11]. The SAP is satisfied provided that, for all $\mathbf{u}^h$ on the fine grid, there exists $\mathbf{v}^{2h}$ on the coarse grid such that

$$||\mathbf{u}^h - P\mathbf{v}^{2h}||^2_{A^h} \leq \frac{C}{||A^h||}||A^h\mathbf{u}^h||^2, \tag{2}$$

where $||\cdot||$ is the discrete L^2 norm. To relate this bound back to the accuracy of the discretization, first assume that A^h comes from a first-order finite element discretization of an H^2-regular elliptic partial differential equation, $Lu = f$, with $f \in L^2$. Then $||A^h|| \leq c/h^2$ for some constant, $c > 0$, and $||A^h\mathbf{u}^h|| = ||f^h|| = ||Qf|| \leq ||Q|| \cdot ||f|| \leq \beta$ for some constant, $\beta > 0$, where Q is an interpolation operator from L^2 to grid h (note that Q interpolates between continuous and discrete function spaces and is assumed to be bounded). Also note that the Galerkin condition implies that $\mathbf{u}^{2h}$, the solution of the coarse-grid system $A^{2h}\mathbf{u}^{2h} = P^T\mathbf{f}^h$, provides an optimal approximation to the fine-grid solution, $\mathbf{u}^h$, in the A^h-norm [6]. Thus, putting everything together yields

$$||\mathbf{u}^h - P\mathbf{u}^{2h}||^2_{A^h} \leq ||\mathbf{u}^h - P\mathbf{v}^{2h}||^2_{A^h} \leq \frac{C}{||A^h||}||A^h\mathbf{u}^h||^2 \leq (C\beta^2/c)h^2.$$

Thus, (1) is confirmed for first-order discretizations by assuming the SAP. This provides an algebraic framework for analyzing discretization-accuracy convergence of FAMG.

4. Standard FAMG Convergence. When working in a geometric setting, it is natural to construct interpolation such that assumption (1) is satisfied. In an algebraic setting, however, enforcing this bound or an equivalent SAP as outlined above can be much more difficult. Standard AMG hierarchies often do not exhibit these approximation properties, preventing FAMG from obtaining the discretization-accuracy convergence that FMG often achieves.

To demonstrate the above claims about FMG and FAMG, consider the model problem of a Poisson equation discretized on a square with bilinear finite elements:

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

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 2 shows convergence of the relative total error,

$$\text{Relative Error} = \frac{||\mathbf{v}_i^h - \mathbf{u}||}{||\mathbf{u}||},$$

for FMG vs. FAMG, where $\mathbf{u}$ is the true solution evaluated on the fine grid and $\mathbf{v}_i^h$ is the solution obtained by the nested iteration process (solving on each coarse grid using a single V-cycle, then projecting up) plus i additional V-cycles on the fine grid. This discussion uses the discrete L^2 norm throughout. Note that the error shown stalls at the level of discretization accuracy since it is measured against an analytic solution, $\mathbf{u}$. As the grid is refined (and 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. Notice, however, that FMG achieves accuracy on the order of the discretization with one (or at most two) fine-grid V-cycle, independent of problem size.

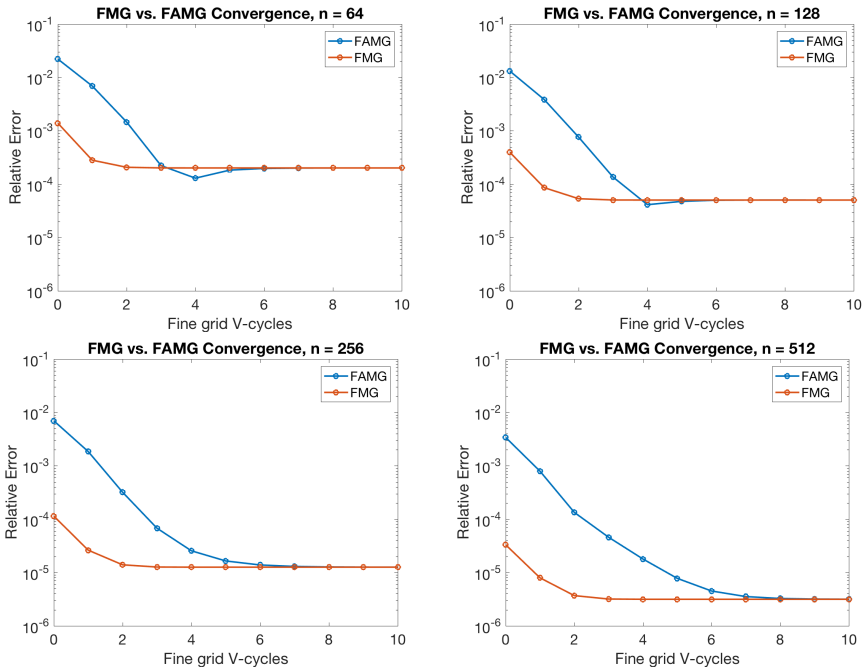

FIG. 2. *Relative total error convergence for standard FAMG vs. FMG on $n \times n$ finite element grids of increasing size.*

For the model problem with the appropriate coloring scheme (i.e. a full coarsening, C/F splitting), the coarse grids generated in the AMG hierarchy are the same as the coarse grids generated by GMG. The interpolation operators between 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. This subtle difference in interpolation is enough, however, to prevent AMG from satisfying assumption (1). Figure 3 shows the interpolation error, $||\mathbf{u}^h - P\mathbf{u}^{2h}||$, on each level (with 0 representing the finest level) for the GMG and AMG hierarchies. These errors are calculated with the manufactured right-hand side, f, specified above. GMG obtains exactly $O(h^2)$ convergence of the interpolation error (which is to be expected for the given first-order discretization), while AMG fails this requirement. This indicates that the standard AMG hierarchy also does not achieve the SAP for this problem. The SAP can be numerically calculated by computing C on each level of the AMG hierarchy such that

$$C = ||A^h|| \cdot ||(A^h)^{-1} - P(P^T A^h P)^{-1} P^T|| \geq ||A^h|| \max_{\mathbf{u}^h} \min_{\mathbf{u}^{2h}} \frac{||\mathbf{u}^h - P\mathbf{u}^{2h}||^2_{A^h}}{||A^h \mathbf{u}^h||^2}. \tag{3}$$

This value for C is shown across the levels of the AMG hierarchy in Figure 3. The growth in C for finer grids indicates that C is not constant, but grows with the grid size, implying that the AMG hierarchy does not achieve the SAP. Thus, even though the difference in interpolation operators between the AMG and GMG methods is subtle (again, they differ only in the way they choose weights near the boundary for the model problem), they exhibit very different behavior across the levels of the hierarchy.

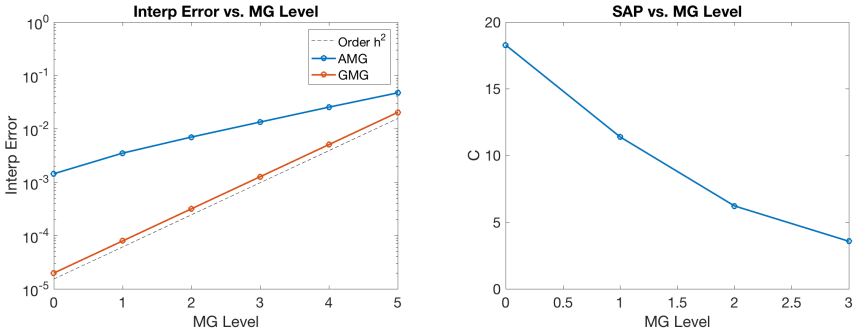

FIG. 3. *Interpolation error across levels of the multigrid hierarchies generated by standard AMG vs. GMG (with 0 representing the finest level), and the value of C as defined in (3) across multigrid levels. Note that a smaller fine grid was used to calculate C (since this is expensive to calculate directly), resulting in fewer multigrid levels than shown for the plot of interpolation error.*

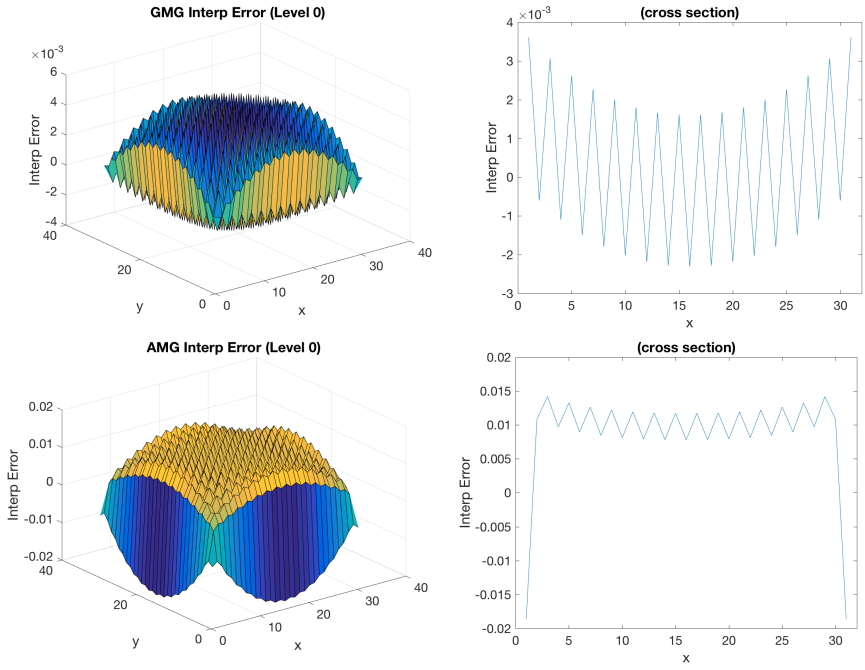

FIG. 4. *Plot of the finest-level interpolation error over the grid for GMG (top) vs. standard AMG (bottom). The cross section is taken through the middle of the domain.*

A visualization of the error, $\mathbf{u}^h - P\mathbf{u}^{2h}$, over the domain is helpful in better understanding the nature of interpolation error for GMG vs. AMG. As seen in Figure 4, the interpolation error for GMG is small and oscillatory. This is precisely the sort of error that is effectively removed by a V-cycle on the fine grid. For AMG, however, there is a large smooth component to the error produced by large discrepancies near the boundary, which pollute the entire domain. More V-cycles are required to remove this smooth mode in the error. Thus, the above analysis suggests that an FAMG cycle might be improved by changing AMG interpolation in some way to recover $O(h^2)$ scaling of the interpolation error.

5. AMG Interpolation Error. Throughout this paper, "standard" AMG refers to the Ruge-Stüben method for generating an AMG hierarchy [4, 8]. The basic principle of any multigrid scheme is that of the complimentary process of fine-grid relaxation and coarse-grid correction. AMG assumes a relaxation scheme (such as weighted Jacobi or Gauss-Seidel) for which the slowly converging error components have relatively small residuals. So the slowly decaying error, $\mathbf{e}$, which should be addressed by the coarse-grid correction, loosely yields $A^h\mathbf{e} \approx \mathbf{0}$. This motivates AMG's choice of interpolation. Rewriting the assumption $A^h\mathbf{e} \approx \mathbf{0}$ componentwise and separating the strongly connected coarse points from other connections yields

$$a_{ii}e_i \approx -\sum_{j\in C_i} a_{ij}e_j - \sum_{j\in D_i} a_{ij}e_j,$$

where the a_{ij}'s are entries from the matrix A^h, C_i is the set of strongly connected coarse points, and D_i is the set of remaining connections. To obtain an interpolation formula from the above heuristic, assuming that the goal is to interpolate point i from all strongly connected coarse-grid points, the contributions from D_i must be collapsed either to point i or to points in C_i so that e_i is calculated as a weighted sum of only the interpolary coarse-grid values:

$$e_i = \sum_{j\in C_i} w_{ij}e_j.$$

The w_{ij}'s above are referred to as interpolation weights and become the entries in the interpolation matrix, P. Standard AMG collapses weakly connected points to point i and strongly connected fine points to C_i to obtain the interpolation weights:

$$w_{ij} = -\frac{a_{ij} + \sum\limits_{k\in D_i^s} a_{ik}\frac{a_{kj}}{\sum\limits_{j'\in C_i} a_{kj'}}}{a_{ii} + \sum\limits_{m\in D_i^w} a_{im}}, \tag{4}$$

where D_i^s is the set of strongly connected fine-grid points and D_i^w is the set of weakly connected points. Note that AMG collapses the fine-grid connections such that the constant vector is interpolated exactly: $P\mathbf{1}^{2h} = \mathbf{1}^h$. With interpolation defined, the remaining pieces of an AMG hierarchy follow from the usual variational property and Galerkin condition: restriction is defined as the transpose of interpolation, $R = P^T$, and the coarse-grid operators are formed via $A^{2h} = P^TA^hP$.

6. Improving AMG Interpolation. In a GMG hierarchy, linear interpolation between grids can be explicitly enforced, guaranteeing $O(h^2)$ scaling of the interpolation error. In an algebraic setting, there is no notion of geometry and, as such, no way to explicitly enforce linear interpolation. Thus, it may be necessary to leverage algebraic information in order to emulate linear interpolation as much as possible.

In an algebraic setting, the main approach to constructing interpolation is to ensure that the appropriate vectors lie in the range of interpolation on each level. As mentioned in the previous section, the standard Ruge-Stüben way of choosing interpolation weights in AMG ensures that the constant vector is in the range. The analysis of the Section 3 shows that this property is not sufficient for achieving good

interpolation near the boundary. Thus, the range of interpolation must be corrected or enriched in some way to recover good interpolation everywhere in the domain.

One approach to improving the range of interpolation is to modify the way in which fine-grid connections are collapsed to coarse-grid points when forming interpolation. Standard AMG seeks to preserve the constant vector when collapsing these connections, but this approach may be modified to fit an arbitrary vector $\mathbf{x}$, resulting in the following formula for interpolation weights [5]:

$$w_{ij} = -\frac{a_{ij} + \sum_{k \in D_i^s} a_{ik} \frac{a_{kj} x_k}{\sum_{j' \in C_i} a_{kj'} x_{j'}}}{a_{ii} + \sum_{m \in D_i^w} a_{im}}. \tag{5}$$

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

In general, the coarse grid should correct components of the error that are in the near-kernel of A^h. For the model Poisson problem, the eigenvectors are known, so let $\mathbf{x}$ be the eigenvector associated with the smallest eigenvalue (the sine hump) and define interpolation according to (5). As shown in Figure 5, defining interpolation in this way restores $O(h^2)$ 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-kernel of any given operator is generally not known a priori. A fully adaptive method like the one described in [5] could be used to find the near-kernel, but these methods have expensive setup costs.

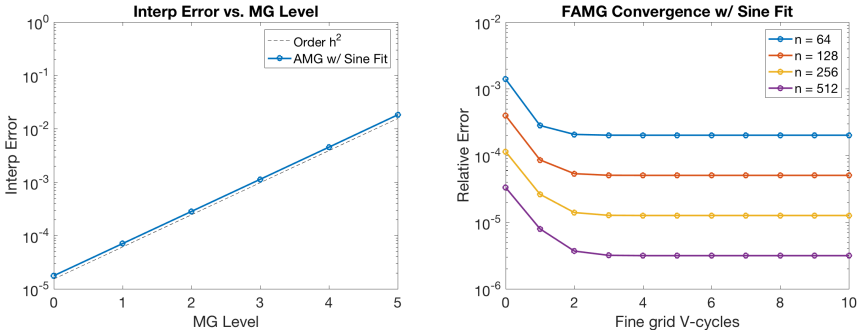

FIG. 5. *Interpolation error across multigrid levels and FAMG convergence when choosing interpolation to fit the sine hump.*

A more efficient approach here is to use the idea of trying to fit the local kernel at each node, that is, choose a different vector, $\mathbf{x}$, at each point to satisfy $(A^h\mathbf{x})_i = 0$. Throughout the rest of this section, "boundary nodes" refer to nodes that are directly adjacent to the boundary of the domain (the actual nodes on the boundary of the domain are assumed to be eliminated from the system because of the Dirichlet conditions there), and the remaining points are referred to as "interior nodes." Note that nodes are still identified as boundary or interior solely through algebraic information, specifically by the row sum of the operator, A^h at that node. For the model problem, the operator, A^h, has row sum zero for all interior nodes, so letting $\mathbf{x} = \mathbf{1}$ satisfies $(A^h\mathbf{x})_i = 0$ for all interior nodes, i. For the boundary nodes, however, the operator does not have row sum zero, and the constant is no longer in the local kernel. A better approach is to fit a constant vector that has been smoothed. Thus, interpolation is

constructed on each level using (5) as the formula for the weights and choosing which vector, $\mathbf{x}$, to use at each point, i, according to

$$\mathbf{x} = \begin{cases} \mathbf{1}, & i \in I, \\ (M^{-1})^{\nu}\mathbf{1}, & i \in B, \end{cases} \tag{6}$$

where I is the set of all interior nodes, B is the set of boundary nodes, and $(M^{-1})^{\nu}$ represents ν applications of some smoother, M^{-1}. This has the effect of modifying interpolation only near the boundary, which is appropriate. Recall from Section 3 that the boundary is, in fact, the only place where interpolation weights differ from geometric multigrid for the model problem.

As shown in Figure 6, this method also restores $O(h^2)$ 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 used at the boundary requires more smoothing iterations in order for the method to perform well (that is, ν grows with problem size). When $n = 64$ (where the fine grid consists of $n \times n$ finite elements), two Jacobi iterations were sufficient, whereas for $n = 512$, eight Jacobi iterations were required to obtain good FAMG convergence. Figure 6 shows the effect on interpolation error for different values of ν.

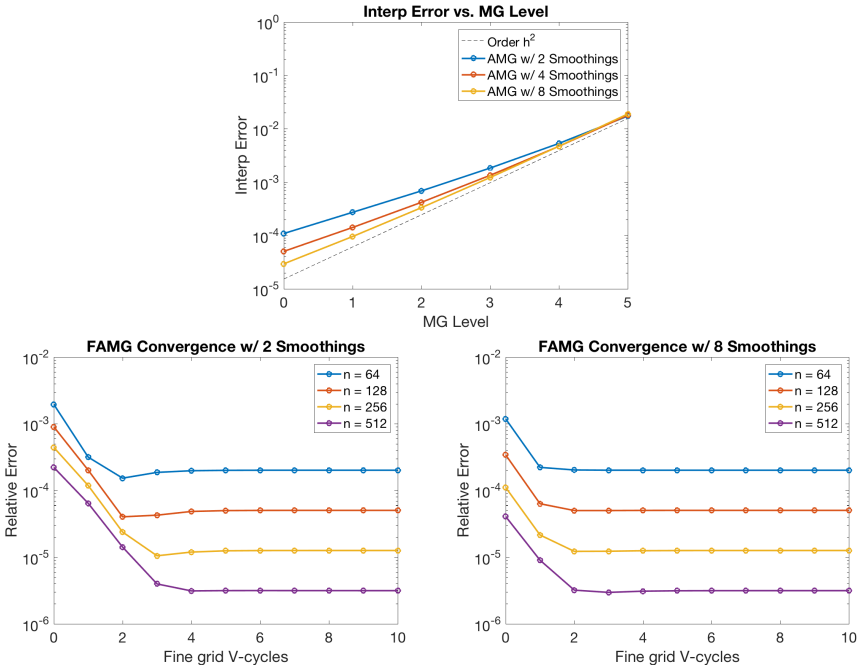

FIG. 6. *Interpolation error across multigrid levels and FAMG convergence when choosing interpolation to fit* $\mathbf{x}$ *as defined in (6). Convergence is shown when using* $\nu = 2$ *smoothings and when using* $\nu = 8$ *smoothings. Interpolation error is shown for* $\nu = 2, 4, 8$ *smoothings.*

Taking the ideas presented above one step further, a much better choice for $\mathbf{x}$ may be obtained by doing a local smoothing only on the boundary points (where again a point is determined to be on the boundary if its row sum is nonzero). That is, rather than applying a global smoothing M^{-1} on a constant vector, relax only

when a point's row sum is nonzero, leaving other points unchanged. Denote this boundary-only smoother as M_B^{-1}. For the model problem, performing $\mathbf{x} = (M_B^{-1})^\nu \mathbf{1}$ converges in only $\nu = 2$ boundary-Gauss-Seidel relaxations (independent of n) to a vector $\mathbf{x}$ that closely approximates linear interpolation to the Dirichlet boundaries of the model problem. Constructing interpolation using formula (5) and replacing (6) with

$$\mathbf{x} = \begin{cases} \mathbf{1}, & i \in I, \\ (M_B^{-1})^\nu \mathbf{1}, & i \in B, \end{cases} \tag{7}$$

yields a much more scalable method for obtaining good interpolation error. For the model problem, performing only $\nu = 2$ applications of M_B^{-1} was sufficient to obtain the $O(h^2)$ interpolation error and good FAMG convergence shown in Figure 7.

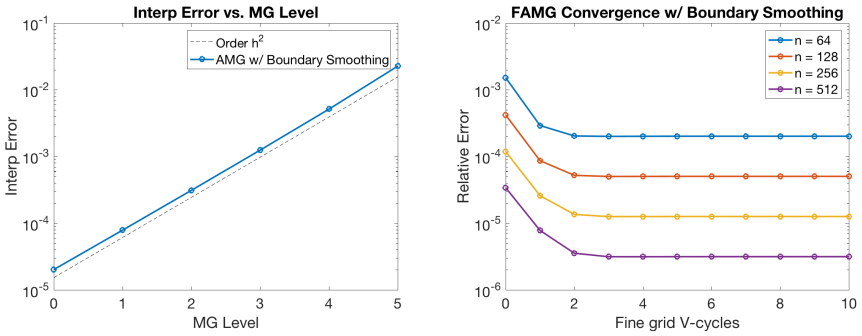

FIG. 7. *Interpolation error across multigrid levels and FAMG convergence when choosing interpolation to fit* $\mathbf{x}$ *as defined in (7) using* $\nu = 2$ *boundary smoothings.*

The boundary smoothing technique can be used to successfully generate AMG interpolation that achieves $O(h^2)$ interpolation error for a variety of problems. Figure 8 shows proper interpolation error scaling across multigrid levels and good FAMG performance when using boundary smoothing for a jump-coefficient Poisson problem with constant right-hand side, $f = 1$:

$$\begin{aligned} -\nabla \cdot (q \nabla u) &= f \,,\ u \in \Omega = [-1,1] \times [-1,1], \\ u &= 0 \,,\ u \in \partial\Omega, \\ q(x,y) &= \begin{cases} 1 \\ 1000 \end{cases} . \end{aligned}$$

An even more compelling example where AMG interpolation with boundary smoothing works well is the linear elasticity problem with constant right-hand side, $f = 200$:

$$\begin{aligned} -\nabla \cdot (\sigma(\mathbf{u})) &= f, \\ \sigma(\mathbf{u}) &= (\nabla \cdot \mathbf{u}) I + (\nabla \mathbf{u} + \nabla \mathbf{u}^T). \end{aligned}$$

Results for this problem in Figure 9 indicate that the boundary smoothing technique may be successfully applied to systems of equations by combining it with the

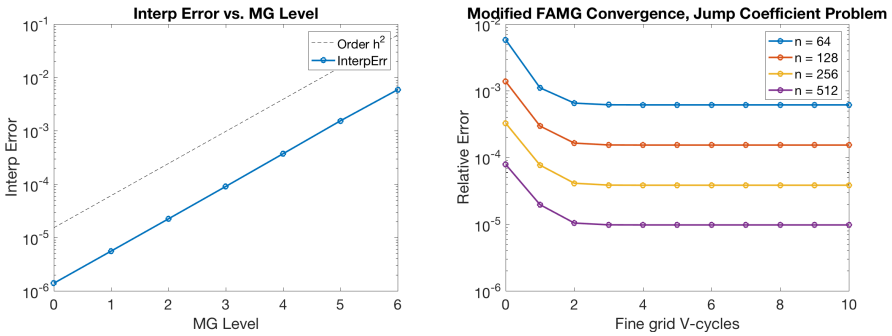

FIG. 8. *Interpolation error across multigrid levels and FAMG convergence when choosing interpolation to fit* $\mathbf{x}$ *as defined in (7) using* $\nu = 2$ *boundary smoothings for the jump coefficient problem.*

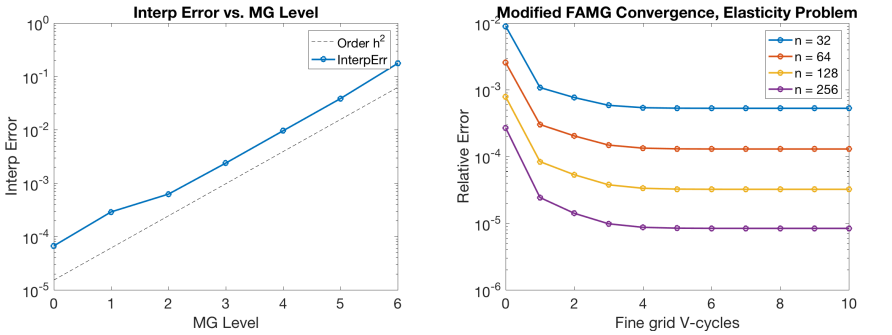

FIG. 9. *Interpolation error across multigrid levels and FAMG convergence when choosing interpolation to fit* $\mathbf{x}$ *as defined in (7) using* $\nu = 2$ *boundary smoothings for the linear elasticity problem.*

"unknown-based" approach for constructing AMG interpolation for systems, in which interpolation is constructed separately for each variable in a block fashion [8].

It should be noted that for both the jump-coefficient and elasticity problems, standard AMG interpolation does not scale appropriately, resulting in poor FAMG convergence. Thus, the boundary smoothing technique provides a meaningful fix for both of these problems. Another important consideration, however, is that both problems used regular coarse grids and homogeneous Dirichlet boundary conditions. Modifying the boundary conditions or using irregular coarse grids (as are often generated by AMG in general) result in worse performance for FAMG using interpolation constructed by boundary smoothing.

In order to provide some additional, heuristic understanding of the boundary smoothing process, one can think of this method as solving a one-dimensional subproblem around the boundary of the domain. Let A_I, A_{IB}, A_{BI}, and A_B be the matrices describing the interior-interior, interior-boundary, boundary-interior, and boundary-boundary connections in the matrix A^h respectively, then we may rewrite $A^h\mathbf{x} = \mathbf{0}$ in block form as:

$$\begin{bmatrix} A_I & A_{IB} \\ A_{BI} & A_B \end{bmatrix} \begin{bmatrix} x_I \\ x_B \end{bmatrix} = \mathbf{0}, \tag{8}$$

where x_I are the interior degrees of freedom and x_B are the boundary degrees of freedom of the vector $\mathbf{x}$. Note that this matrix equation has only the trivial solution when A^h is non-singular, however, if we first set $\mathbf{x} = \mathbf{1}$, then the first block of equations is satisfied:

$$\begin{bmatrix} A_I & A_{IB} \end{bmatrix} \begin{bmatrix} x_I \\ x_B \end{bmatrix} = \mathbf{0},$$

because $\mathbf{1}$ is in the local kernel for all interior nodes. Now, the bottom block of equations may similarly be used to determine an appropriate $\mathbf{x}$ for the boundary by fixing $x_I = \mathbf{1}$ and solving for x_B in

$$A_B x_B = -A_{BI}\mathbf{1}. \tag{9}$$

The boundary smoothing technique, $\mathbf{x} = (M_B^{-1})^\nu \mathbf{1}$, converges to the solution of (9). Writing things in this way exposes some of the guiding principles for constructing the modified interpolation described in this section. Finding a global near-kernel vector, $\mathbf{x}$, and constructing interpolation to fit this vector is effective but also expensive. For some problems, it can be much cheaper to find multiple vectors, $\mathbf{x}$, which fit local kernels over certain subsets of the domain (like the interior or the boundary), and constructing interpolation based on these vectors also yields good results.

7. Computational Cost of FAMG vs. AMG V-cycles. To quantify the difference in computational cost between FAMG and AMG V-cycles, first define a work unit (WU) as the cost of applying the operator A^h on the finest grid. So one WU has cost $O(n)$, where n is the size of the finest grid. The cost of all the operations performed during a multigrid cycle may then be estimated in terms of WUs. A single V(1,1) cycle performs two relaxations (2 WU), a residual calculation (1 WU), restriction, and interpolation (each less than 1 WU) on the finest grid for a total cost less than 5 WU. These operations are then repeated on the coarse grids. Assuming a coarsening factor of 4 (as in the two-dimensional model problem), the total cost can then be estimated by summing over the levels:

$$(5\text{WU}) * \sum_{i=0}^{\log_4(n)-1} \left(\frac{1}{4}\right)^i \leq (5\text{WU}) * \left(\frac{1}{1-\frac{1}{4}}\right) = \frac{20}{3}\text{WU}.$$

This estimate of the cost for a V-cycle can then be used to estimate the cost of an FAMG cycle. Note that a V-cycle on grid $2h$ is about $(1/4)$ the cost of a V-cycle on grid h (again assuming a coarsening factor of 4). So, if a single V(1,1) cycle is used to solve on each level during FAMG, then the total cost is

$$\left(\frac{20}{3}\text{WU}\right) * \sum_{i=0}^{\log_4(n)-1} \left(\frac{1}{4}\right)^i \leq \left(\frac{20}{3}\text{WU}\right) * \left(\frac{1}{1-\frac{1}{4}}\right) = \frac{80}{9}\text{WU}.$$

Note that a single FAMG cycle is only slightly more expensive than a single V-cycle, but delivers convergence to discretization accuracy in one step. V-cycles often exhibit some constant convergence factor, ρ, independent of n. The amount of error

reduction required to achieve discretization accuracy for an order p discretization of a two-dimensional problem is $O(n^{-p/2})$, so the number of V-cycles required to achieve discretization accuracy is

$$\nu = \log_{1/\rho}(n^{p/2}) = (p/2)\log_{1/\rho}(n) = O(\log(n)).$$

This means that the total computational effort required to achieve discretization accuracy using V-cycles scales like $O(n\log(n))$ as oppose to FAMG's $O(n)$ scaling.

Timing results observed for the model problem verify the scalings described above and also demonstrate an overall speedup in solve time for FAMG over V-cycles, as shown in Figure 10. Solve times for FAMG show the amount of time taken to perform a single FAMG cycle (including a single V-cycle on the fine grid). Solve times for V-cycles show the amount of time taken to perform enough V-cycles on a random initial guess to reduce the relative total error to a level less than 1.5 times the relative total error achieved by FAMG. Note that the FAMG cycle here uses the modified interpolation described in the previous section (utilizing boundary relaxation to modify the vector fit at the boundary) and, thus, achieves the level of discretization accuracy in a single cycle. Additional V-cycles were required to achieve similar accuracy as problem size increased, resulting in the expected $O(n\log(n))$ scaling. The V-cycles here use a standard AMG hierarchy, though the modified AMG hierarchy yields similar V-cycle performance.

Setup times are also included in Figure 10. The results shown demonstrate that constructing the modified AMG hierarchy described in the previous section (using boundary smoothing to generate a vector to fit at the boundaries) has nearly the same cost as constructing a standard hierarchy. The only added cost when constructing the modified hierarchy comes from performing the boundary relaxations. This cost is $O(\sqrt{n})$ and relatively small compared to the overall cost of a standard hierarchy setup.

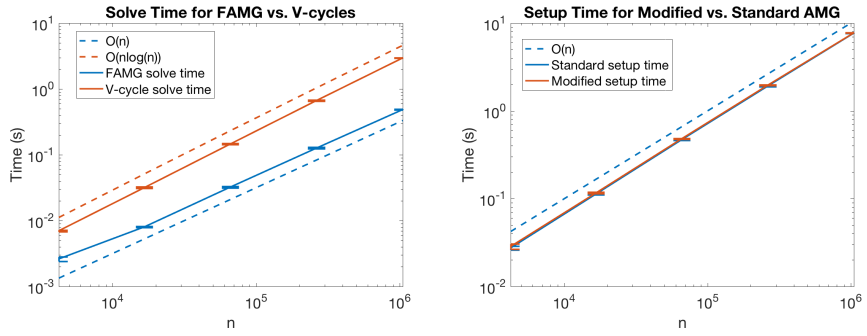

FIG. 10. *Timing results verifying the $O(n)$ scaling for the solve time of FAMG, the $O(n\log(n))$ scaling for the solve time of V-cycles, and the $O(n)$ scaling for the setup time of both standard and modified AMG hierarchies. 95% confidence intervals are shown (calculated over 20 different timing runs), though these intervals are very tight.*

8. Conclusions. Applying a full multigrid cycle to a standard AMG hierarchy does not generally yield a solution at the level of discretization accuracy in a single cycle even when AMG V-cycles yield good convergence factors. Analysis of a model Poisson problem shows that the interpolation operators produced by AMG

are insufficient to achieve the desired discretization-accuracy convergence. Several different approaches were described for improving AMG interpolation in order to recover convergence to discretization accuracy. Fully adaptive methods, which place the near-kernel of the operators in the range of interpolation on each level, should be successful, but these methods have expensive setup costs. Focusing instead on fitting the local kernels for the operator and modifying interpolation to treat points with zero row sum differently from points with nonzero row sum shows promise as a scalable, efficient method for recovering good FAMG convergence. This paper demonstrates that such an approach works very well on some simple tests problems, including jump-coefficient Poisson and linear elasticity, and reduces the computational cost of solving this problem to discretization accuracy using AMG from $O(n \log(n))$ to $O(n)$.

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. Also, the discussion here has focused on first-order discretizations. For higher-order discretizations, the problem of constructing interpolation sufficient to recover good FAMG convergence becomes more complicated. Future work will address both of these topics.

REFERENCES

[1] M. F. Adams, J. Brown, M. Knepley, and R. Samtaney, *Segmental Refinement: A Multigrid Technique for Data Locality*, arXiv.org, (2014), arXiv:1406.7808v3 [math.NA].

[2] R. Bank, R. Falgout, T. Jones, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid domain and range decomposition (amg-dd/amg-rd)*, SIAM Journal of Scientific Computing, 37 (2015), pp. S113–S136.

[3] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Mathematics of Computation, 31 (1977), pp. 333–390.

[4] A. Brandt, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid (amg) for sparse matrix equations*, Sparsity and Its Applications, (1984).

[5] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive algebraic multigrid*, SIAM Journal of Scientific Computing, 27 (2005), pp. 1261–1286.

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

[7] C. Rodrigo, F. J. Gaspar, C. W. Oosterlee, and I. Yavneh, *Accuracy Measures and Fourier Analysis for the Full Multigrid Algorithm*, SIAM Journal on Scientific Computing, 32 (2010), pp. 3108–3129.

[8] J. Ruge and K. Stüben, *Algebraic multigrid (amg)*, Frontiers in Applied Mathematics, 3 (1987), pp. 73–130.

[9] A. Thekale, T. Gradl, K. Klamroth, and U. Rüde, *Optimizing the number of multigrid cycles in the full multigrid algorithm*, Numerical Linear Algebra with Applications, 17 (2010), pp. 199–210.

[10] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2000.

[11] P. Vassilevski, *Multilevel Block Factorization Preconditioners: Matrix-based Analysis and Algorithms for Solving Finite Element Equations*, Springer, 2008.