

NON-GALERKIN COARSE GRIDS FOR ALGEBRAIC MULTIGRID*

ROBERT D. FALGOUT[†] AND JACOB B. SCHRODER[†]

Abstract. Algebraic multigrid (AMG) is a popular and effective solver for systems of linear equations that arise from discretized partial differential equations. While AMG has been effectively implemented on large scale parallel machines, challenges remain, especially when moving to exascale. In particular, stencil sizes (the number of nonzeros in a row) tend to increase further down in the coarse grid hierarchy, and this growth leads to more communication. Thus, as problem size increases and the number of levels in the hierarchy grows, the overall efficiency of the parallel AMG method decreases, sometimes dramatically. This growth in stencil size is due to the standard Galerkin coarse grid operator, P^TAP , where P is the prolongation (i.e., interpolation) operator. For example, the coarse grid stencil size for a simple three-dimensional (3D) seven-point finite differencing approximation to diffusion can increase into the thousands on present day machines, causing an associated increase in communication costs. We therefore consider algebraically truncating coarse grid stencils to obtain a non-Galerkin coarse grid. First, the sparsity pattern of the non-Galerkin coarse grid is determined by employing a heuristic minimal “safe” pattern together with strength-of-connection ideas. Second, the nonzero entries are determined by collapsing the stencils in the Galerkin operator using traditional AMG techniques. The result is a reduction in coarse grid stencil size, overall operator complexity, and parallel AMG solve phase times.

Key words. multigrid, algebraic multigrid, non-Galerkin coarse grid, parallel computing, high-performance computing

AMS subject classifications. 65F08, 65N55

DOI. 10.1137/130931539

1. Introduction. The goal of this paper is to introduce a non-Galerkin coarse grid strategy that improves the parallel performance of algebraic multigrid (AMG) [6, 14]. AMG is of interest because it is a popular and effective solver of large sparse linear systems in parallel, by virtue of being *scalable* or *optimal* and solving a linear system with n unknowns with only $O(n)$ work. Consider solving the linear system

$$(1.1) \quad A\mathbf{x} = \mathbf{b},$$

where $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$. For the problems considered here, $A \in \mathbb{R}^{n \times n}$ is assumed to be a symmetric and positive definite (SPD) matrix.

There are two main components to a multigrid method: the *smoother* (or *relaxation* method) and the *coarse grid correction* step. The coarse grid correction step involves operators that transfer information between the fine and coarse “grids,” denoted more generally by the space \mathbb{R}^n and the lower dimensional (coarse) vector space \mathbb{R}^{n_c} . We focus on classical Ruge–Stüben style AMG (RS-AMG) [14], which constructs the coarse space using a disjoint F/C splitting of the fine grid unknowns, with the

*Submitted to the journal’s Software and High-Performance Computing section August 2, 2013; accepted for publication (in revised form) March 20, 2014; published electronically June 26, 2014. This work was performed by an employee of the U.S. Government or under U.S. Government contract. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344 (LLNL-JRNL-641635).

<http://www.siam.org/journals/sisc/36-3/93153.html>

[†]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551 (falgout2@llnl.gov, schroder2@llnl.gov).

Algorithm 1: Twogrid($A, \mathbf{x}, \mathbf{b}$).

1	Input: A	{Matrix}
2	x	{Initial guess to linear system}
3	b	{Right-hand-side vector}
4	Relax ν_1 times on $A\mathbf{x} = \mathbf{b}$	
5	Correct $\mathbf{x} \leftarrow \mathbf{x} + P(P^TAP)^{-1}P^T(\mathbf{b} - A\mathbf{x})$	
6	Relax ν_2 times on $A\mathbf{x} = \mathbf{b}$	
7	return x	

C -points forming the coarse grid. We let $P : \mathbb{R}^{n_c} \rightarrow \mathbb{R}^n$ be the *interpolation* (or *prolongation*) operator, and generally the *restriction* operator is taken to be P^T . Optimality is achieved when smoothing and coarse grid correction are complementary. In the classical setting of scalar elliptic problems, this means that the smoother is a simple iterative algorithm such as Gauss–Seidel, which is effective at reducing high frequency error. The remaining low frequency error is then accurately represented and efficiently eliminated on coarser grids via the coarse grid correction step. As a result, all error frequencies are uniformly damped with a linear cost. The two-grid multigrid method for solving (1.1) is then defined in Algorithm 1. In practice, we use a multilevel method to solve (1.1) by recursively applying Algorithm 1 to the correction step in line 5. The proposed work considers replacing the Galerkin coarse grid operator P^TAP with a sparser approximation A_c , which maintains the AMG convergence rate but improves parallel efficiency.

Controlling the sparsity pattern of P^TAP is important because AMG (including the target BoomerAMG package in *hypre*) produces coarse grids with a maximum coarse grid stencil size that grows with problem size. This is problematic in parallel, because as stencil size increases on coarser levels, standard matrix partitioning couples processors that were not coupled on previous, finer levels. Thus, the communication pattern also grows as problem size increases, with the risk that for exascale problems, the maximum coarse grid stencil size may produce unscalable communication patterns. This increase in communication is well documented [16, 8]. In particular, the work [8] describes how even the best current practices in *hypre* still result in a sharp increase in number of messages and in overall communication cost when moving to coarse grids. For example, the model diffusion problem considered in [8] resulted in a growth from 6 MPI sends on the finest level to 245 MPI sends on the worst coarse grid for a machine with 65K cores. The result was that the time spent on some coarse levels in the hierarchy, despite being much smaller in terms of the number of nonzeros and the number of unknowns, was actually larger than the time spent on the finest level.

The need to improve coarse grid sparsity has been a main driver for research-based improvements to *hypre*. Classical parallel AMG (i.e., classical modified interpolation and Falgout coarsening [9]) is indeed a useful and powerful method, but at large numbers of cores, the communication cost ruins what is otherwise a computationally optimal method. As an example, consider Figure 1, where the time to solution for classical parallel AMG is plotted for the simple three-dimensional (3D) seven-point finite-difference diffusion operator. A weak scaling study is done with 25,000 unknowns per core on an Intel cluster with a fast InfiniBand QDR interconnect, and the time grows dramatically with core count. This result led to the important developments of aggressive coarsening [16] and extended interpolation [15], which lead in

this experiment to a reduction in coarse grid stencil size from the thousands to the hundreds. The resulting time to solution is also commensurately reduced, in large part due to reduced communication, as indicated by the “Best practices AMG”¹ plot in Figure 1. However, even with these advances, the time to solution is still growing in the figure, leaving us room for improvement through communication reducing strategies.

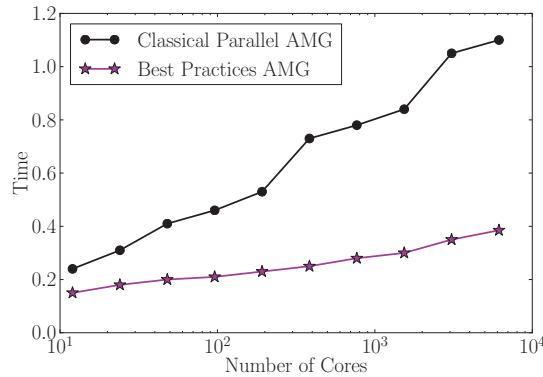


FIG. 1. *AMG weak scaling as motivation: Parallel timings for classical parallel AMG and current best practices for a simple model diffusion problem.*

Our strategy is to further reduce the communication costs by replacing P^TAP with a sparser approximation. While we focus on a purely algebraic approach, non-Galerkin methods have already been explored in settings where geometric information is used to aid the method [2, 20] in choosing sparsity patterns and matrix coefficients. This previous work already indicates that a significant reduction in coarse grid stencil size is possible without a serious reduction in multigrid convergence. The work [5] discusses, but does not experiment with, a purely algebraic non-Galerkin coarse grid approach that computes each coarse grid matrix row based on minimizing a quadratic functional. While this is an intriguing approach, it has the goal of directly computing coarse grid equations (i.e., coarse grid matrix rows) without the use of P . This is a significantly more difficult and expensive proposition than the proposed approach. Our goal is simpler because we continue to use P to define a Galerkin coarse grid and then sparsify this coarse grid to derive a more efficient multigrid scheme. More recent work [17] has followed a path similar to ours in sparsifying coarse grid Galerkin operators for smoothed aggregation (SA) [19], and while not fully developed or tested in parallel, shows much promise.

Our overall strategy is to begin with an existing AMG method and to remove entries from the Galerkin operator each time a new level in the hierarchy is formed. Thus our goal is to have sparser coarse grid operators than given by Galerkin coarse grids and to also maintain AMG convergence when compared to the corresponding Galerkin coarse grid method. The algorithm consists of two phases. The first phase chooses a sparsity pattern for the new coarse grid operator, and the second phase removes or “collapses” entries in P^TAP that lie outside of that sparsity pattern. While we focus on the BoomerAMG package in *hypre*, we believe that the approach

¹For a complete discussion of the *hypre* parameters used for “Best practices AMG” and “Classical parallel AMG,” see section 5.3.

is general enough to be applicable to most AMG codes. To this end, we have also implemented the proposed method in an SA setting (see section 5.2).

In section 2, we first provide a mathematical motivation for our approach. In section 3, we state the resulting algorithm. In section 4, we describe a near-optimal non-Galerkin coarse grid operator and compare our method to the near-optimal method for model diffusion problems. In section 5, we give serial results for a variety of diffusion and elasticity problems, followed by parallel results for diffusion problems.

2. Mathematical motivation. In this section, we provide the mathematical motivation for our approach. Assume that an AMG method is applied to an SPD matrix, A . During construction of an AMG hierarchy, the Galerkin coarse grid, $A_g = P^T A P$, has been computed, but the rows contain too many nonzeros, or the number of nonzeros per row is growing too quickly. Our goal is to replace A_g with a sparser approximation, A_c , that is spectrally equivalent in the sense that

$$(2.1) \quad \exists \alpha, \beta \in \mathbb{R}^+ \text{ such that } \alpha \leq \lambda(A_c^{-1} A_g) \leq \beta,$$

with α and β both close to 1. Alternatively, we show in Theorem 2.1 that minimizing

$$(2.2) \quad \theta \equiv \|I - A_c A_g^{-1}\|_2$$

is a sufficient heuristic for defining A_c to produce an effective AMG method or preconditioner.

Building on results in [11], define the symmetric two-grid error propagator for AMG as

$$(2.3) \quad E_K = (I - M^{-T} A)^\nu (I - P K^{-1} P^T A) (I - M^{-1} A)^\nu,$$

where K is the coarse grid matrix and M defines the relaxation method (e.g., the diagonal of A in the case of Jacobi). Let E_{TG} denote the error propagator for the standard Galerkin case, $K = A_g$, and let E_{PTG} denote the propagator for the *perturbed* two-grid case, $K = A_c$. Next, define preconditioners B_{TG} and B_{PTG} induced by the equations

$$(2.4a) \quad E_{TG} = I - B_{TG}^{-1} A,$$

$$(2.4b) \quad E_{PTG} = I - B_{PTG}^{-1} A.$$

We now relate the spectral radius $\rho(E_{PTG})$ of the perturbed method to that of the Galerkin method. Similarly, we bound the condition number $\kappa(B_{PTG}^{-1} A)$ in terms of $\kappa(B_{TG}^{-1} A)$.

THEOREM 2.1. *Define E_{TG} , E_{PTG} , B_{TG} , B_{PTG} , and θ as above, and assume that A_g and A_c are SPD. If $\theta < 1$, then*

$$(2.5) \quad \kappa(B_{PTG}^{-1} A) \leq \left(\frac{1 + \theta}{1 - \theta} \right) \kappa(B_{TG}^{-1} A)$$

and

$$(2.6) \quad \rho(E_{PTG}) \leq \max \left(\lambda_{\max}(B_{TG}^{-1} A) \cdot \frac{1}{1 - \theta} - 1, 1 - \lambda_{\min}(B_{TG}^{-1} A) \cdot \frac{1}{1 + \theta} \right).$$

Proof. Using a result from [11], we have that

$$(2.7) \quad \begin{aligned} \lambda_{\max}(B_{PTG}^{-1} A) &\leq \lambda_{\max}(B_{TG}^{-1} A) \cdot \max(\lambda_{\max}(A_c^{-1} A_g), 1) \\ &= \lambda_{\max}(B_{TG}^{-1} A) \cdot \frac{1}{\min(\lambda_{\min}(A_g^{-1} A_c), 1)}, \end{aligned}$$

and similarly

$$(2.8) \quad \lambda_{\min}(B_{PTG}^{-1}A) \geq \lambda_{\min}(B_{TG}^{-1}A) \cdot \frac{1}{\max(\lambda_{\max}(A_g^{-1}A_c), 1)}.$$

Since A_g and A_c are SPD, we have that

$$\begin{aligned} \rho(I - A_g^{-1}A_c) &= \max(\lambda_{\max}(A_g^{-1}A_c) - 1, 1 - \lambda_{\min}(A_g^{-1}A_c)) \\ &\leq \|I - A_g^{-1}A_c\|_2 = \|I - A_cA_g^{-1}\|_2 = \theta. \end{aligned}$$

Hence, $\lambda_{\max}(A_g^{-1}A_c) \leq 1 + \theta$, $\lambda_{\min}(A_g^{-1}A_c) \geq 1 - \theta$, and (2.5) follows from (2.7) and (2.8). Since A and B_{PTG} are SPD, (2.6) follows from the additional fact that

$$\rho(E_{PTG}) = \max(\lambda_{\max}(B_{PTG}^{-1}A) - 1, 1 - \lambda_{\min}(B_{PTG}^{-1}A)). \quad \square$$

2.1. Heuristics. In this section, we further motivate the algorithm heuristics based on Theorem 2.1. Our basic goal is to construct an operator A_c with a given sparsity pattern such that θ is small. To state this more rigorously, we first introduce some notation. Define a *matrix nonzero pattern* (or *sparsity pattern*) as a set of tuples $\{(i, j)\}$, and denote the space of matrices with given sparsity pattern \mathcal{N} by

$$(2.9) \quad \mathcal{N} \equiv \{A \in \mathbb{R}^{n \times n} : a_{ij} \neq 0 \text{ only if } (i, j) \in \mathcal{N}\}.$$

Similarly, define a *vector nonzero pattern* as a set of indices $\{i\}$, and denote the space of vectors with given nonzero pattern \mathcal{I} by

$$(2.10) \quad \mathcal{I} \equiv \{\mathbf{x} \in \mathbb{R}^n : x_j \neq 0 \text{ only if } j \in \mathcal{I}\}.$$

Let \mathcal{N}_c and \mathcal{N}_g represent the sparsity patterns of A_c and A_g , respectively. Then, our aim is to define $\mathcal{N}_c \subseteq \mathcal{N}_g$ and $A_c \in \mathcal{N}_c$ such that θ is small. We begin by stating A_c as a perturbation of A_g ; i.e., we let $A_c \in \mathcal{N}_c$, $A_g \in \mathcal{N}_g$, and

$$(2.11a) \quad A_c = A_g + E,$$

$$(2.11b) \quad \|I - A_cA_g^{-1}\|_2 = \|EA_g^{-1}\|_2 = \theta.$$

Next, we develop heuristics for (2.11b) intended to reduce θ . First, we develop a heuristic for “midrange” or high energy modes, \mathbf{x} ,

$$(2.12a) \quad A_g\mathbf{x} = \lambda\mathbf{x},$$

where $\|\mathbf{x}\|_2 = 1$ and λ is not small, say, $\lambda \in [\rho(A_g)/10, \rho(A_g)]$. Our simple row-wise heuristic uses Gershgorin rings by enforcing

$$(2.12b) \quad \|\mathbf{e}_i\|_1 \leq \gamma \|\mathbf{a}_i^g\|_1,$$

where $(\mathbf{e}_i)^T$ and $(\mathbf{a}_i^g)^T$ represent the i th row of E and A_g , and $\gamma \in [0, 1]$. Thus letting $\max_i \|\mathbf{a}_i^g\|_1 = k\rho(A_g)$, we have

$$(2.12c) \quad \rho(E) \leq \gamma \max_i \|\mathbf{a}_i^g\|_1,$$

$$(2.12d) \quad \rho(E) \leq k\gamma \rho(A_g).$$

Finally, for the mode \mathbf{x} in question, we can say

$$(2.12e) \quad \|EA_g^{-1}\mathbf{x}\|_2 = \frac{1}{\lambda} \|E\mathbf{x}\|_2 \leq \frac{\rho(E)}{\lambda} \leq k\gamma \frac{\rho(A_g)}{\lambda}.$$

For typical AMG problems from standard discretizations of the Poisson operator, k is small—usually in the vicinity of 2 or 3. Thus for midrange or high energy modes, the impact on θ from (2.12e) is minimal for the γ chosen here, which are on the order of 0.01.

Much of the preceding argumentation for midrange to high energy modes centers around global eigenvalues, but it is important that the heuristic (2.12b) is row-wise. This locality implies that enforcing (2.12b) for a specific γ targets the locally midrange to high energy modes, which may or may not correspond to the global midrange to high energy modes.

The other case concerns \mathbf{x} as a low energy eigenmode such that $\|\mathbf{x}\|_2 = 1$, but $A_g\mathbf{x} \approx \mathbf{0}$. Then, θ is small when

$$(2.13a) \quad E\mathbf{x} \approx \mathbf{0} \quad \text{if} \quad A_g\mathbf{x} \approx \mathbf{0}.$$

Since AMG typically makes a priori assumptions about the local behavior of the near null space, we can leverage that here as well target (2.13a) is a heuristic. In particular, we enforce accuracy for a set of vectors, B , that represent the near null space, i.e.,

$$(2.13b) \quad A_c B = A_g B \quad \Leftrightarrow \quad EB = \mathbf{0}.$$

The standard choice for B is the constant, but it could also be problem-dependent, e.g., the rigid-body modes for elasticity. For the most common case of $B = \mathbf{1}$, we guarantee (2.13b) with classical AMG inspired stencil collapsing to remove the unwanted entries in A_g , so that $E\mathbf{1} = \mathbf{0}$. Classical stencil collapsing was originally developed for constructing interpolation operators and is well known to be accurate for near null space modes. However, AMG interpolation techniques do not explicitly address accuracy for higher energy modes, thus necessitating heuristic (2.12b). When B contains multiple vectors, a generalized collapsing strategy is used, which we detail later in section 3.3.1.

While this approach is based on the above heuristics, it should be noted that much of the successful classical AMG framework, e.g., strength-of-connection and interpolation formulas, is based on similar heuristic assumptions. Put another way, if the constant does not adequately represent a low energy vector \mathbf{x} locally, then the classical AMG framework in which we operate will already be problematic.

3. Algorithm. In this section, we first describe our algorithm for finding a suitable sparsity pattern \mathcal{N}_c for A_c . Second, we describe the algorithm for eliminating entries in A_g based on \mathcal{N}_c for the case of classical AMG scalar problems. The strategy is based on classical AMG stencil collapsing. Third, we discuss one possible generalization of our algorithm to other PDEs, such as linearized elasticity, where the near null space of the matrix contains more than the constant vector.

The heuristics in section 2 guide our algorithmic choices. The process of finding the sparsity pattern utilizes a drop tolerance that guarantees satisfaction of equation (2.12b), which is the heuristic targeting accuracy in A_c for midrange to high energy modes. The process of stencil collapsing to eliminate entries is then done in such a way to target the second heuristic (2.13b) and give A_c accuracy for the important near null space modes.

3.1. Choosing the sparsity pattern. The goal of the sparsity pattern choice is to ensure an adequate nonzero pattern for spectral equivalence between A_g and A_c . We approach this problem in a two-phase manner. The first phase initializes the sparsity pattern using the matrix graph of P and the fine grid operator A to target sufficient connections in A_c to approximate the Galerkin matrix stencil. The second phase guarantees heuristic (2.12b) to more directly target spectral equivalence for midrange to high energy modes.

Minimal sparsity pattern. Considering the first phase, we must at a minimum choose coarse grid sparsity patterns that *allow* for reproducing a good approximation to the Galerkin matrix stencil. A central difficulty is the fact that it is not known how to define a sufficient minimal sparsity pattern on the coarse grid for a scalable AMG method. Because of this fact, our minimal sparsity pattern relies on heuristics.

We begin by noting that non-Galerkin strategies regularly succeed for geometric multigrid and result in bounded stencil growth. In contrast to AMG, stencils typically grow because distance-three (or farther) connections on the fine grid become distance-one connections on the coarse grid. To target eliminating these coarse grid connections that correspond to longer distance fine grid connections, we choose

$$(3.1) \quad \hat{\mathcal{N}}_c = \{(i, j) \text{ such that } (P_I^T A P + P^T A P_I)_{ij} \neq 0\}$$

as the minimal sparsity pattern, where P_I refers to injection between the coarse and fine grids. Typically P will have a connectivity pattern based on A , so it is easy to see that $P_I^T A P$ will only connect distance-one and distance-two fine grid unknowns on the coarse grid. This results in a sparsity pattern similar to those achieved by geometric multigrid schemes.

This initial pattern allows for important matrix entries in A_g with a small magnitude to be preserved on coarse grids in $\hat{\mathcal{N}}_c$. These entries would normally be dropped if the classical strength-of-connection measure (which is based on magnitude) were used to compute $\hat{\mathcal{N}}_c$. Instead (3.1) uses the information (often of a geometric nature) present in the matrix graph of A to maintain connections on the coarse grid between all fine grid distance-one C -point connections and some fine grid distance-two C -point connections. (This assumes that classical RS-AMG type interpolation and coarsening schemes are used.) So while we will later use a collapsing scheme based on RS-AMG strength-of-connection and interpolation to eliminate entries in A_g , the concept of strength-of-connection is not sufficient by itself to construct \mathcal{N}_c . We note that [17] uses a similar approach in an SA setting to obtain a sparsity pattern.

To illustrate this issue, consider a simple example of grid-aligned anisotropic diffusion, $-u_{xx} - \epsilon u_{yy} = f$, discretized with the standard five-point finite-difference stencil given by A in (3.2a) below. The stencil for A_g in (3.2a) is that yielded by one level of semicoarsening (coarsening in the x -direction only) as obtained with RS-AMG. The stencil for A_c in (3.2b) is a rediscritization of the problem on the coarse grid and provides a suitable non-Galerkin coarse grid for A_g . However, if the connections in the direction of weak diffusion are dropped in A_c to yield \tilde{A}_c , the spectral equivalence bounds are h -dependent. Hence, the value of θ in Theorem 2.1 approaches one, and

we cannot guarantee an h -independent two-grid method.

$$(3.2a) \quad A := \begin{bmatrix} & -\epsilon & \\ -1 & (2+2\epsilon) & -1 \\ & -\epsilon & \end{bmatrix}, \quad A_g := \begin{bmatrix} -\frac{\epsilon}{4} & -\frac{3}{2}\epsilon & -\frac{\epsilon}{4} \\ -\frac{1}{2} + \frac{\epsilon}{2} & (1+3\epsilon) & -\frac{1}{2} + \frac{\epsilon}{2} \\ -\frac{\epsilon}{4} & -\frac{3}{2}\epsilon & -\frac{\epsilon}{4} \end{bmatrix},$$

$$(3.2b) \quad A_c := \begin{bmatrix} & -2\epsilon & \\ -\frac{1}{2} & (1+4\epsilon) & -\frac{1}{2} \\ & -2\epsilon & \end{bmatrix}, \quad \tilde{A}_c := \begin{bmatrix} & 0 & \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ & 0 & \end{bmatrix}.$$

To see this h -dependence, we do a simple derivation using the well-known [7] eigenvectors v_i and eigenvalues λ_i of the classic second-order finite-difference operator with stencil $[-1 \ 2 \ -1]$. Letting this one-dimensional stencil be defined on a grid with n points, the corresponding eigenvalues and eigenvectors of A_c defined on an $n \times n$ grid are given by

$$(3.3a) \quad \lambda_{ij} = \frac{1}{2}\lambda_i + 2\epsilon\lambda_j, \quad v_{ij} = v_i \otimes v_j,$$

where the indexing $i, j < n$. The spectrum of \tilde{A}_c can similarly be expressed with

$$(3.3b) \quad \tilde{A}_c v_{ij} = \frac{1}{2}\lambda_i v_{ij}.$$

Using this knowledge of the spectra, we have

$$(3.3c) \quad \lambda(\tilde{A}_c^{-1} A_c) = \frac{\frac{1}{2}\lambda_i + 2\epsilon\lambda_j}{\frac{1}{2}\lambda_i} = 1 + 4\epsilon\lambda_i^{-1}\lambda_j.$$

Hence,

$$(3.3d) \quad 1 \leq \lambda(\tilde{A}_c^{-1} A_c) \lesssim 1 + 4\epsilon \left(\frac{4-h}{h} \right).$$

In summary, h -independence for θ is achieved when the closest small magnitude entries in the direction of weak diffusion are included in the stencil for A_c . We avoid the h -dependent sparsity pattern of \tilde{A}_c by using $\hat{\mathcal{N}}_c$ from (3.1) as the minimal sparsity pattern, which for this example is

$$(3.4) \quad \hat{\mathcal{N}}_c := \begin{bmatrix} & * & \\ * & * & * \\ & * & \end{bmatrix}.$$

The use of this heuristic minimal sparsity pattern is critical because the high and low energy heuristics (2.12b) and (2.13a) do not prevent the sparsity pattern of \tilde{A}_c from being chosen. Moreover, the classical concept of strength-of-connection cannot find a suitable sparsity pattern here, and if it were employed to determine $\hat{\mathcal{N}}_c$ based on A_g , then the pattern for \tilde{A}_c would be chosen. On the other hand, it is important that the minimal pattern $\hat{\mathcal{N}}_c$ is restrictive enough to exclude connections in A_g that represent longer distance connections on the fine grid.

Algorithm 2: ComputeSparsity(A_g, P, P_I).

```

1 Input:  $A_g$                                 {Galerkin operator}
2        $P$                                 {Interpolation operator}
3        $P_I$                              {Injection operator}
4
5  $\mathcal{N}_c \leftarrow \emptyset$ 
6 for  $(i, j)$  such that  $a_{ij}^g \neq 0$  do
7    $\mathcal{N}_c \leftarrow \mathcal{N}_c \cup \{(i, j)\}$       {Initial pattern}
8
9 for  $i = 1$  to  $nrows(A_g)$  do
10   Initialize set  $\mathcal{K}$ :  $\mathcal{K}_m$  is index of  $m$ th smallest off-diagonal nonzero in row  $i$ 
11   for  $m = 1$  to  $|\mathcal{K}|$  do
12     if  $2 \sum_{j \notin \mathcal{N}_{ci} \setminus \mathcal{K}_m} |a_{ij}^g| \leq \gamma \sum_j |a_{ij}^g|$     {Equation (2.12b) satisfied}
13     then
14        $\mathcal{N}_{ci} \leftarrow \mathcal{N}_{ci} \setminus \mathcal{K}_m$       {Remove entry  $\mathcal{K}_m$ }
15
16 for  $(i, j)$  such that  $(P_I^T A P + P^T A P_I)_{ij} \neq 0$  do
17    $\mathcal{N}_c \leftarrow \mathcal{N}_c \cup \{(i, j)\}$       {Union with minimal pattern}
18 return  $\mathcal{N}_c$ 

```

Improving the sparsity pattern. Next, we improve $\hat{\mathcal{N}}_c$ by enforcing the heuristic (2.12b) through the following procedure, in order to target accuracy for midrange to high energy modes. Let the set of neighbors of i in \mathcal{N}_c (i.e., the allowed nonzeros in row i) be

$$\mathcal{N}_{ci} = \{j, \text{ such that the tuple } (i, j) \in \mathcal{N}_c\}.$$

The procedure then begins by initializing the pattern \mathcal{N}_c with the pattern of A_g . Entries are then removed from \mathcal{N}_c , starting with the smallest in magnitude, until any further removal of entries would violate

$$(3.5) \quad 2 \sum_{j \notin \mathcal{N}_{ci}} |a_{ij}^g| \leq \gamma \sum_j |a_{ij}^g|.$$

Assuming classical stencil collapsing will be done to compute final matrix entry values (as outlined in section 3.2), this procedure guarantees (2.12b) for the final A_c , even before the stencil collapsing step. The factor of 2 essentially represents the change made to A_g when dropping an entry, plus the maximum change to A_g possible when collapsing that entry to the allowed nonzero entries. Last, we ensure the minimal sparsity pattern from (3.1) by taking the union of \mathcal{N}_c and $\hat{\mathcal{N}}_c$; i.e., we replace \mathcal{N}_c with $\mathcal{N}_c \cup \hat{\mathcal{N}}_c$. Algorithm 2 gives a detailed description of this process. If a symmetric pattern is desired, \mathcal{N}_c can be symmetrized, such that if $(i, j) \in \mathcal{N}_c$, then (j, i) is added to the set \mathcal{N}_c as well.

3.2. Eliminating entries in A_g for scalar problems. The goal when eliminating entries in A_g to obtain A_c is to not change the action on the near null space of A_g . We rely on the sparsity pattern choice from section 3.1 to avoid changing the action on other modes. We eliminate entries in A_g based on a classical AMG stencil collapsing approach, because this approach has been used for decades [6, 14, 18] to

develop interpolation formulas which are accurate for the constant-based near null space modes critical for classical scalar problems. In other words, this lumping procedure is (roughly speaking) as effective at preserving the near null space as classical AMG interpolation is accurate for the near null space.

The dropping strategy begins by initializing A_c as a copy of A_g . Then, each $a_{ij}^c \neq 0$ such that $(i, j) \notin \mathcal{N}_c$ is eliminated from A_c . The stencil collapsing algorithm eliminates a_{ij}^c by adding a fraction of that entry to each of j 's strongly connected neighbors in row i . The fractional lumping is done such that the row sum does not change and the local constant-based near null space is thus unchanged. We begin by letting S be the strength-of-connection matrix defined by the classical AMG measure with respect to A_g , and define the neighbors of j in S as

$$(3.6a) \quad \mathcal{N}_{sj} = \{k, \text{ such that } s_{jk} \neq 0\}.$$

Next, we find U , which represents the strong connections of j shared by the nonzero pattern of row i , that is,

$$(3.6b) \quad U = \mathcal{N}_{sj} \cap \mathcal{N}_{ci}.$$

This set U represents the unknowns to which a_{ij}^c will be lumped. If strong neighbors are found ($U \neq \emptyset$), we avoid changing the diagonal and remove i from U . This avoids changing the center of each row's Gershgorin disc, if possible. If the diagonal entry of the matrix is changed too much, then the scale of that equation will subsequently change, and we have observed this being detrimental to performance. We then compute the fractions by which the entry will be lumped and carry out the fractional lumping such that the near null space is preserved and $E\mathbf{1} = 0$:

$$(3.6c) \quad \sigma = \sum_{k \in U} |s_{jk}|,$$

$$(3.6d) \quad a_{ik}^c \leftarrow a_{ik}^c + (|s_{jk}|/\sigma) a_{ij}^c \quad \text{for } k \in U.$$

Similar to classical stencil collapsing, if no strong neighbors are found ($U = \emptyset$), the entry is lumped to the diagonal. After all entries are eliminated, the matrix A_c may be symmetrized,

$$(3.6e) \quad A_c \leftarrow 0.5(A_c^T + A_c),$$

followed by diagonal lumping to preserve the row sum,

$$(3.6f) \quad a_{ii}^c \leftarrow a_{ii}^c + \sum_j a_{ij}^g - \sum_j a_{ij}^c.$$

Algorithm 3 describes in detail this stencil collapsing algorithm. In section 3.3 we show how this procedure produces a collapsing operator identical to that used by classical AMG interpolation strategies.

While the stencil collapsing process preserves the row sum of the operator, the preservation of symmetry in line 20 changes this row sum. Thus, line 22 is required to modify the diagonal and preserve the row sum. The values added to the diagonal in line 22 are typically small, but they are necessary for maintaining accuracy for the near null space of the operator. From a different point of view, this symmetrization

Algorithm 3: NonGalerkin(A_g, S, \mathcal{N}_c).

```

1 Input:  $A_g$  {Galerkin operator}
2  $S$  {Strength-of-connection operator based on  $A_g$ }
3  $\mathcal{N}_c$  {Desired sparsity pattern}
4
5  $A_c \leftarrow A_g$  {Initialize with Galerkin coarse grid}
6 for  $i = 1$  to  $nrows(A_c)$  do
7   for  $j$  such that  $a_{ij} \neq 0$  do
8     if  $j \notin \mathcal{N}_{ci}$  then
9        $U \leftarrow \mathcal{N}_{sj} \cap \mathcal{N}_{ci}$ 
10      if  $U = \emptyset$  or  $U = \{i\}$  then
11         $a_{ii}^c \leftarrow a_{ii}^c + a_{ij}^c$  {No neighbors, lump to diagonal}
12      else
13         $U \leftarrow U \setminus \{i\}$  {Avoid changing diagonal}
14         $\sigma = \sum_{k \in U} |s_{jk}|$ 
15        for  $k \in U$  do
16           $a_{ik}^c \leftarrow a_{ik}^c + (|s_{jk}|/\sigma) a_{ij}^c$  {Lump to neighbors}
17         $a_{ij}^c \leftarrow 0$ 
18      else
19        Do nothing
20  $A_c \leftarrow 0.5(A_c^T + A_c)$  {Symmetrize}
21 for  $i = 1$  to  $nrows(A_c)$  do
22    $a_{ii}^c \leftarrow a_{ii}^c + \sum_j a_{ij}^g - \sum_j a_{ij}^c$  {Preserve row sum}
23 return  $A_c$ 

```

step can be thought of as a way of symmetrically dropping entries, where lines 15 and 16 are replaced with

$$(3.7a) \quad \text{for } k \in U \text{ do } a_{ik}^c \leftarrow a_{ik}^c + (1/2)(|s_{jk}|/\sigma)a_{ij}^c,$$

$$(3.7b) \quad a_{ki}^c \leftarrow a_{ki}^c + (1/2)(|s_{jk}|/\sigma)a_{ij}^c,$$

followed by a fix-up step that lumps to the diagonal to preserve the row sum.

Remark 1. The symmetrization steps (3.7a) and (3.7b) disturb the property (2.12b), which had been guaranteed with (3.5). We have not observed in our experiments that the symmetrization step changes (3.5) significantly. So, while the property (2.12b) could be guaranteed by another processing step after symmetrization, we have not found it to be needed.

Remark 2. We lump entries to the diagonal in line 11 when no strong neighbors are found ($U = \emptyset$). However, these entries may also be left in the matrix. In our experiments, few entries were lumped to the diagonal, so the impact on complexity is small. This question remains a topic of active research.

Remark 3. The stencil collapsing approach of Algorithms 2 and 3 does preserve definiteness for M-matrices. An M-matrix is a common multigrid operator where the diagonal is positive, all off-diagonals are nonpositive, and row sums are nonnegative. Thus by a Gershgorin disc argument, M-matrices are at least positive semidefinite. When the proposed approach operates on an M-matrix, all off-diagonals remain non-positive (off-diagonals are changed only by adding a negative value); the diagonal

remains positive; and the row sum is unchanged. Thus the resulting A_c is still an M-matrix and hence definite. If the symmetrization step is done, then A_c is SPD. If the interpolation method chosen preserves the M-matrix property on coarse grids, then the proposed method will still yield an SPD preconditioner. However, for general matrices, there is no way to guarantee that the proposed method preserves definiteness. Avoiding a loss of definiteness is a topic of active research, but we note that in our experiments accelerating GMRES and CG with the method led to similar results, implying that no significant loss of definiteness occurred for the test problems considered.

In summary, this algorithm removes entries from a coarse grid operator in a way that preserves the spectrum for constant-like near null space modes. The algorithm relies on the choice of \mathcal{N}_c to guarantee that the resulting coarse grid is accurate for midrange to high energy modes. Additionally, the algorithm relies primarily on local row-wise computations that require little communication, making the method well suited for the parallel setting.

3.3. Eliminating entries in A_g in a general setting. In this section, we consider the case where the matrix requires stencil collapsing that preserves multiple modes in B ; i.e., we must enforce heuristic (2.13b) when B has multiple columns. The motivation is to extend the proposed method to problems such as elasticity, where the near null space consists of multiple vectors, e.g., the rigid-body modes. We first present classical stencil collapsing in terms of the general approach and then discuss one way to extend this perspective to the multiple-mode case.

Consider Figure 2, where the matrix graph relative only to unknown 1 is depicted, and a local ordering is used. In the picture, unknowns 5 and 6 are to be eliminated from the stencil and collapsed to unknowns 2, 3, and 4. The gray dotted lines and associated weights represent strength-of-connection couplings between 5 and 6 relative to their neighbors 2, 3, and 4. The strength-of-connection couplings have been normalized such that the two connections for unknown 5 sum to 1, and likewise for unknown 6.

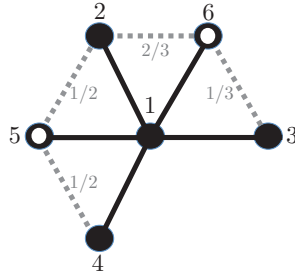


FIG. 2. *Stencil collapsing example.*

Let the set of off-diagonal unknowns that are maintained in row $i = 1$ be denoted by

$$(3.8) \quad \mathcal{N}_{ci} = \{2, 3, 4\},$$

and let the set of off-diagonal unknowns to be eliminated be denoted by

$$(3.9) \quad \chi_i = \mathcal{N}_{gi} \setminus \mathcal{N}_{ci} = \{5, 6\}.$$

Then the corresponding generalization for classical stencil collapsing can be represented with a local interpolation-like operator G_i that operates only on nonzero entries, i.e.,

$$(3.10a) \quad \begin{bmatrix} a_{ii}^c & (\mathbf{a}_i^c|_{\mathcal{N}_{ci}})^T \end{bmatrix} = \begin{bmatrix} a_{ii}^g & (\mathbf{a}_i^g|_{\mathcal{N}_{ci}})^T & (\mathbf{a}_i^g|_{\chi_i})^T \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & I \\ 0 & G_i \end{bmatrix},$$

where $\mathbf{a}_i^c|_{\mathcal{N}_{ci}}$ is the vector of off-diagonal entries after stencil collapsing, $a_{ii}^c = a_{ii}^g$, and

$$(3.10b) \quad G_i = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 2/3 & 1/3 & 0 \end{bmatrix}.$$

Note that all other entries in row i of A_c are zero. The entry $(G_i)_{0k}$ represents the normalized strength-of-connection entry between unknowns 5 and k , and $(G_i)_{1k}$ likewise corresponds to unknown 6. More precisely, recall the definition of $U \leftarrow \mathcal{N}_{sj} \cap \mathcal{N}_{ci}$ for each row i from Algorithm 3, and define G_i as

$$(3.11a) \quad G_i \in \mathbb{R}^{|\chi_i| \times |\mathcal{N}_{ci}|},$$

$$(3.11b) \quad (G_i)_{uv} = \frac{s_{(\chi_i)_u, (\mathcal{N}_{ci})_v}}{\sum_{k \in U} |s_{(\chi_i)_u, k}|}.$$

In summary, the operator G_i interpolates from unknowns in \mathcal{N}_{ci} to unknowns in χ_i using normalized strength-of-connection values that interpolate between strongly connected unknowns and preserve near null space mode(s). For this example, all row sums of G_i equal 1. In general, classical stencil collapsing will always result in a row sum of 1 in G_i .

Remark 4. Equation (3.10a) defines the general case of stencil collapsing, but it does not allow for entries to be lumped to the diagonal as in Algorithm 3. This can be observed because the first column is all zero, except for the (1,1) entry. This was an engineering choice meant to protect the diagonal entry from large changes during the process of fitting multiple modes into the span of G_i , which can cause relatively large changes to entries. For the case when an a_{ij} is being eliminated and this results in $U = \{i\}$ or $U = \emptyset$, then that a_{ij} is not eliminated and remains in the matrix. Lumping this entry to the diagonal is an option and would result in a small change, but we choose to maintain a consistent approach here and leave the diagonal unchanged in all cases. This does not change the resulting algorithmic behavior because very few such a_{ij} are traversed in our test cases.

On the other hand, in Algorithm 3, the changes to the diagonal are guaranteed to be small through the choice of γ . So, we allow these small changes in order to drop entries as aggressively as possible.

3.3.1. Accommodating multiple near null space modes. We now present the generalized stencil collapsing case, where G_i is postprocessed to accommodate multiple near null space modes. This set of vectors B is usually specified on the finest level and then transferred to each coarse level with the restriction operator $P_I^T B$. Multigrid methods typically require a set of such modes for construction of interpolation. Even classical AMG methods implicitly assume $B = \mathbf{1}$ by virtue of the interpolation formulas which are explicitly designed to preserve the constant.

Our strategy updates the G_i from classical stencil collapsing in a least-squares fashion so that G_i interpolates multiple vectors B accurately. The intention is that

Algorithm 4: Generalized_NonGalerkin(A_g, S, \mathcal{N}_c, B).

```

1 Input:  $A_g$  {Galerkin operator}
2  $S$  {Strength-of-connection operator based on  $A_g$ }
3  $\mathcal{N}_c$  {Desired sparsity pattern}
4  $B$  {Near null space modes for  $A_g$ , ordered in importance}
5
6  $A_c \leftarrow A_g$  {Initialize with Galerkin coarse grid}
7 for  $i = 1$  to  $nrows(A_c)$  do
8    $k_i \leftarrow \max\left(\left\lfloor \frac{|\mathcal{N}_{ci}|}{2} \right\rfloor - 2, 1\right)$  {Use first  $k_i$  columns of  $B$ }
9   Compute  $G_i$  {Equation (3.11)}
10   $F \leftarrow B|_{\chi_i} - G_i B|_{\mathcal{N}_{ci}}$  {Compute update to  $G_i$ }
11  for  $j = 1$  to  $nrows(G_i)$  do
12    Let  $\mathbf{f}$  and  $\mathbf{g}$  be row  $j$  of  $F$  and  $G_i$ , respectively.
13     $\mathbf{u} \leftarrow \arg\min_{\mathbf{u}} \|(B|_{\mathcal{N}_{ci}})^T \mathbf{u}^T - \mathbf{f}^T\|_2^2$ 
14     $\mathbf{g} \leftarrow \mathbf{g} + \mathbf{u}$ 
15
16    
$$\begin{bmatrix} a_{ii}^c & (\mathbf{a}_i^c|_{\mathcal{N}_{ci}})^T \end{bmatrix} = \begin{bmatrix} a_{ii}^g & (\mathbf{a}_i^g|_{\mathcal{N}_{ci}})^T & (\mathbf{a}_i^g|_{\chi_i})^T \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & I \\ 0 & G_i \end{bmatrix}$$
 {Equation (3.10a)}
17     $\mathbf{a}_i^c|_{\chi_i} \leftarrow 0$  {Zero out entries not in pattern}
17 return  $A_c$ 

```

the minimal update results in a G_i with weights that still favor strong connections. More precisely, we update G_i in a minimal 2-norm sense to fit B and satisfy

$$(3.12a) \quad G_i B|_{\mathcal{N}_{ci}} = B|_{\chi_i},$$

where

$$(3.12b) \quad B|_{\mathcal{N}_{ci}} = B_{uv} \quad \text{for } u \in \mathcal{N}_{ci} \text{ and } v \in \{0, 1, 2, \dots, ncols(B)\},$$

$$(3.12c) \quad B|_{\chi_i} = B_{uv} \quad \text{for } u \in \chi_i \text{ and } v \in \{0, 1, 2, \dots, ncols(B)\}.$$

This property guarantees that the action of $A_c B = A_g B$. To see how (3.12a) accomplishes this, take (3.10a) and right multiply by

$$(3.13) \quad \begin{bmatrix} \mathbf{b}_i^T \\ B|_{\mathcal{N}_{ci}} \end{bmatrix},$$

using the identity (3.12a). The quantity \mathbf{b}_i^T represents B restricted to the i th row. We now describe in Algorithm 4 the generalized approach to stencil collapsing. For the experiments considered here, the sparsity pattern argument \mathcal{N}_c is generated by Algorithm 2.

In addition to Remark 4, there are some important algorithmic notes to make. When solving PDE systems, such as elasticity, where an unknown j belongs to only one of many PDEs, it is critical to limit the collapsing to occur between unknowns in the same PDE. This can easily be implemented in Algorithm 4 by prefiltering the strength matrix S with

$$(3.14) \quad s_{ij} \leftarrow 0 \quad \text{if} \quad \text{PDE_num}(i) \neq \text{PDE_num}(j),$$

where the function $\text{PDE_num}()$ returns the PDE number (say, 0, 1, or 2) for an unknown. In order to compute the PDE number for a coarse unknown, we utilize the C/F splitting present at each level to find the corresponding fine grid unknown for each coarse grid unknown. Another algorithm note concerns the computation of line 13 where $(B|\mathcal{N}_{ci})^T$ may be ill-conditioned. This system is currently solved through a singular value decomposition (SVD) approach, and determining appropriate singular value cut-offs is a topic of future work.

Remark 5. We have observed issues with overfitting and the computation of G_i . As an example, let G_i have four nonzeros per row, and let B contain four or five vectors; then the process of computing G_i to exactly preserve all modes is ill-conditioned; i.e., small changes to B can result in large changes to G_i . We have observed this phenomenon for the model two-dimensional (2D) rotated anisotropic diffusion problems in the results section, when B equals the constant, linears, and quadratics. Here, the coefficients computed for G_i would sometimes vary by two or three orders of magnitude. The resulting coarse grid stencil became inaccurate, and the overall AMG cycle deteriorated or diverged as a result. As a solution, we take only the first k_i columns of B when computing G_i . It is assumed that B is ordered in importance, so that the first k_i vectors are the most critical ones. For this experiment, the vectors were ordered constant, linears, and quadratics, so that when vectors were discarded from B , they represented a quadratic function. We choose k_i to be significantly smaller than the number of nonzeros per row in G_i , or $k = |\mathcal{N}_{ci}|/2$, where $|\mathcal{N}_{ci}|$ is the number of unknowns in \mathcal{N}_{ci} . Thus, the number of modes in B used to compute G_i can change with i . While this strategy is purely an engineering choice, it is a reasonable number of modes to choose from B to avoid overfitting.

4. Evaluating the accuracy of the stencil collapsing approach. In this section, we describe a near-optimal non-Galerkin coarse grid operator and compare it to our method for model diffusion problems. Let $(\mathbf{a}_i^c)^T$ be the i th row of A_c , and let λ_j and \mathbf{v}_j be the eigenvalues and eigenvectors of A_g . Below, we derive an upper bound for θ in (2.2) in terms of \mathbf{a}_i^c , λ_j , and \mathbf{v}_j . In (4.1d), we write $\mathbf{x} = \sum_j \alpha_j \mathbf{v}_j$, with $\alpha_j \in [0, 1]$ and $\sum_j |\alpha_j| = 1$. In (4.1e), we can reverse the order of the min-max problem because it is quadratic in α_j and a_{ij}^c . A useful upper bound on the minimal

θ is then

$$\begin{aligned}
(4.1a) \quad \theta_{\min}^2 &= \min_{A_c \in \mathcal{N}_c} \|(I - A_c A_g^{-1})\|_2^2 \\
(4.1b) \quad &= \min_{A_c \in \mathcal{N}_c} \max_{\|\mathbf{x}\|_2=1} \|(I - A_c A_g^{-1})\mathbf{x}\|_2^2 \\
(4.1c) \quad &= \min_{A_c \in \mathcal{N}_c} \max_{\|\mathbf{x}\|_2=1} \sum_i ((\mathbf{x})_i - (\mathbf{a}_i^c)^T A_g^{-1} \mathbf{x})^2 \\
(4.1d) \quad &= \min_{A_c \in \mathcal{N}_c} \max_{\|\alpha\|_2=1} \sum_i \left(\sum_j \alpha_j (v_{ij} - \lambda_j^{-1} \mathbf{v}_j^T \mathbf{a}_i^c) \right)^2 \\
(4.1e) \quad &= \max_{\|\alpha\|_2=1} \min_{A_c \in \mathcal{N}_c} \sum_i \left(\sum_j \alpha_j (v_{ij} - \lambda_j^{-1} \mathbf{v}_j^T \mathbf{a}_i^c) \right)^2 \\
(4.1f) \quad &= \max_{\|\alpha\|_2=1} \sum_i \min_{\mathbf{a}_i^c \in \mathcal{N}_{ci}} \left(\sum_j \alpha_j (v_{ij} - \lambda_j^{-1} \mathbf{v}_j^T \mathbf{a}_i^c) \right)^2 \\
(4.1g) \quad &\leq \sum_i \min_{\mathbf{a}_i^c \in \mathcal{N}_{ci}} \sum_j (v_{ij} - \lambda_j^{-1} \mathbf{v}_j^T \mathbf{a}_i^c)^2.
\end{aligned}$$

To compute our near-optimal A_c , we note that the \mathbf{a}_i^c which satisfy (4.1g) and thus ultimately bound θ_{\min} are equivalent to

$$(4.2) \quad \operatorname{argmin}_{\mathbf{a}_i^c \in \mathcal{N}_{ci}} \left\| \begin{bmatrix} \mathbf{v}_{i1} \\ \mathbf{v}_{i2} \\ \vdots \\ \mathbf{v}_{in} \end{bmatrix} - \begin{bmatrix} (1/\lambda_1)(\mathbf{v}_1^T)|_{\mathcal{N}_{ci}} \\ (1/\lambda_2)(\mathbf{v}_2^T)|_{\mathcal{N}_{ci}} \\ \vdots \\ (1/\lambda_n)(\mathbf{v}_n^T)|_{\mathcal{N}_{ci}} \end{bmatrix} (\mathbf{a}_i^c)|_{\mathcal{N}_{ci}} \right\|_2^2.$$

Hence, the near-optimal A_c is simply constructed by solving (4.2) for each row i .

To use this tool, we compared our method to the near-optimal A_c on a variety of model 2D diffusion problems which yielded small enough matrices that the computation of the near-optimal A_c was possible. We asked two questions. One, is the two-level AMG convergence rate significantly better with the near-optimal method when compared to the proposed approach? Two, what sparsity pattern is necessary to achieve a performance equivalent to that of the Galerkin approach when using the near-optimal method?

As an example, consider the Q1 bilinear finite element discretization on a regular structured grid with Dirichlet boundaries of the rotated anisotropic diffusion equation

$$(4.3) \quad -(c^2 + \epsilon s^2)u_{xx} - 2(1 - \epsilon)cs u_{xy} - (\epsilon c^2 + s^2)u_{yy} = f,$$

where $\epsilon = 0.001$, $c = \cos(\varphi)$, $s = \sin(\varphi)$, and φ is the angle of rotation. We choose the angle $\varphi = \pi/4$ (which is a particularly difficult angle for the stencil collapsing approach, on a 15×15 grid). The stencil at the center point on the first coarse grid is given in Figures 3a, 3b, 3c, and 3d for the cases of the Galerkin approach, and for the proposed stencil collapsing approach with $\gamma = 0.03$, $\gamma = 0.1$, and $\gamma = 1.0$. The asymptotic convergence rates for Figures 3a, 3b, 3c, and 3d are 0.23, 0.23, 0.40, and 0.47, respectively. The corresponding convergence rates and stencils for the near-optimal approach are nearly identical and are hence omitted. The one exception

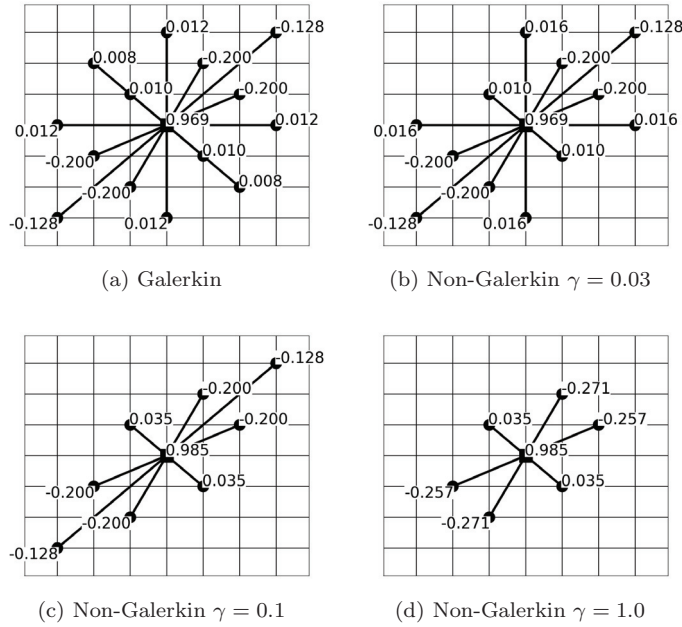


FIG. 3. Center point stencils for first coarse grid operators, comparing Galerkin and stencil collapsing based non-Galerkin for various γ .

was for $\gamma = 1.0$, where the near-optimal approach did yield a better convergence rate and slightly different coefficient values in the cross-stream direction. Thus, the conclusion is twofold. One, the longer distance connections present in Figure 3b, but not present in Figures 3c and 3d, are necessary to regain the performance of the Galerkin approach when using either the near-optimal or stencil collapsing approach. Two, the coefficients computed by stencil collapsing yield a performance similar to that achieved by fitting the eigenvectors in the near-optimal approach.

A number of different angles and finite-difference approximations for isotropic and anisotropic examples were explored with similar results. So, this tool led us to conclude that in general a broader sparsity pattern than $P_I^T A P$ was needed and that $\gamma = 0.03$ yielded such a sparsity pattern for the diffusion problems examined. This tool also allowed us to conclude that the coefficients computed by stencil collapsing are as good as the near-optimal approach for smaller values of γ .

5. Results. In this section, we examine the numerical performance of the method for a variety of diffusion test problems and two 3D elasticity problems. In general, the tests use a V-cycle of RS-AMG to accelerate GMRES with a relative residual tolerance of 10^{-8} . The tests also maintain identical solver parameters and only toggle the non-Galerkin method on and off, so that the standard Galerkin approach and the new method can be fairly compared. The serial tests were done using the PyAMG [4] package, while the parallel tests were done using the *hypr* [10] package, version 2.8.

5.1. Scalar diffusion results. In this section, we examine the proposed non-Galerkin method for scalar diffusion problems. V(2,2) cycles of classical RS-AMG using weighted-Jacobi are used to accelerate GMRES with a relative residual tolerance of 10^{-8} . We choose a V(2,2) cycle with weighted-Jacobi so that our relaxation scheme compares well with the V(1,1) hybrid Gauss-Seidel cycle used in parallel in section

5.3. In particular, weighted-Jacobi parallelizes well, and we choose a V(2,2) cycle because a V(1,1) cycle would exhibit significantly slower AMG convergence than a V(1,1) hybrid Gauss–Seidel cycle for the test problems considered here. The Jacobi weight is $1/\rho(D^{-1}A)$, where D is the diagonal of A .

GMRES is restarted every 15 iterations. The other RS-AMG parameters are direct interpolation, a strength-of-connection tolerance of 0.35, and first and second pass Ruge–Stüben C/F coarsening [14]. For the non-Galerkin algorithm, no post-processing symmetrization of the coarse grid operator is used, because it did not qualitatively change convergence. For all of the test problems in this subsection, including the 3D problems, four consecutive refinements of the initial mesh are considered in order to yield increasing matrix sizes of roughly 16 000, 64 000, 256 000, and 1 000 000.

The tables of results contain the following columns. “Max stencil” reports the maximum stencil size over all levels and is used as a proxy for the amount of communication induced by the AMG hierarchy. The term $\rho_{AMG} = (\|\mathbf{r}_k\|_2 / \|\mathbf{r}_0\|_2)^{1/k}$ refers to the average convergence rate, where \mathbf{r}_k is the final residual value upon convergence after k iterations. “Op. comp.” refers to operator complexity, which is the total number of nonzeros stored in the hierarchy, divided by the number of finest level nonzeros. “Work” refers to work per digit of accuracy, i.e., the amount of work, in fine grid matrix-vector multiplications, required to reduce the residual by one order of magnitude. This quantity is computed with $c/|\log_{10}(\rho_{AMG})|$, where c is the cycle complexity. The cycle complexity is the cost in floating-point operations (relative to one fine grid matrix-vector multiply) to apply one V-cycle. For a V(2,2) cycle, c equals four times the operator complexity because four relaxations are done at each level (two going down and two going up). Work is particularly useful when comparing methods with differing operator complexities because similar iteration counts can represent vastly different amounts of work. A variety of γ values are experimented with, including the case of $\gamma = 1.0$, which is equivalent to using $\tilde{\mathcal{N}}_c$ as the sparsity pattern, and $\gamma = 0.0$, which is equivalent to using the Galerkin coarse grid approach. The overall goal of the tests is to reduce stencil size and hence parallel communication while only negligibly affecting AMG convergence.

In two dimensions, we consider the rotated anisotropic diffusion equation (4.3). The discretizations considered are standard second-order finite differencing and Q1 bilinear finite elements on regular structured grids with Dirichlet boundaries. Experiments are run for a variety of φ values, with worst case angles occurring for roughly $\varphi \approx 2\pi/16$, which is the test case we show in Table 1. While this is a worst case, it is still representative of other angles, in that as γ decreases to $[0.01, 0.03]$, the proposed approach attains similar convergence and nearly identical work units to those of the Galerkin approach ($\gamma = 0.0$). Overall, the non-Galerkin approach is robust for these problems and provides a reduction in operator complexity and stencil size.

The next 2D experiments concern classic coefficient jump problems. We experimented with example problems from [1] and give the results in Table 2 for problem IV from [1]. Here, the diffusion equation is given by

$$(5.1) \quad -\nabla \cdot (d \nabla u) = f,$$

where d is the jumpy coefficient function and the domain is depicted in Figure 4 with Dirichlet boundary conditions. The function d is 1000 in the shaded sawtooth region and 1 elsewhere. The grid is regular, with the coefficient jumps occurring on grid lines. However, the jumps will not be grid-aligned on the coarser grids. The discretization is the classic five-point finite differencing scheme from [1] for coefficient

TABLE 1

2D non-Galerkin results for various γ , rotated anisotropic diffusion by angle of $\varphi = 2\pi/16$, AMG preconditioning GMRES.

γ	Refinement	ρ_{AMG}	Op. comp.	Work	Max stencil
1.0	0	0.28	2.3	17	16
	1	0.44	2.3	25	16
	2	0.51	2.3	32	16
	3	0.56	2.3	36	17
0.03	0	0.31	2.3	17	16
	1	0.39	2.3	23	16
	2	0.46	2.3	28	16
	3	0.50	2.3	31	16
0.01	0	0.28	2.5	19	21
	1	0.31	2.6	21	22
	2	0.35	2.6	23	21
	3	0.37	2.6	24	22
0.0	0	0.26	2.7	19	33
	1	0.28	2.7	20	33
	2	0.30	2.8	21	33
	3	0.32	2.8	23	33

jump problems. The results are similar to the other 2D experiments, in that as γ decreases to $[0.01, 0.03]$, the proposed approach exhibits behavior similar to that of the Galerkin approach. Overall, the non-Galerkin approach is robust for these problems but does not provide much reduction in operator complexity and stencil size. However, this is not surprising, because the original Galerkin approach already exhibited modest complexities and stencil size growth. The main purpose of this experiment is to show the robustness of the method.

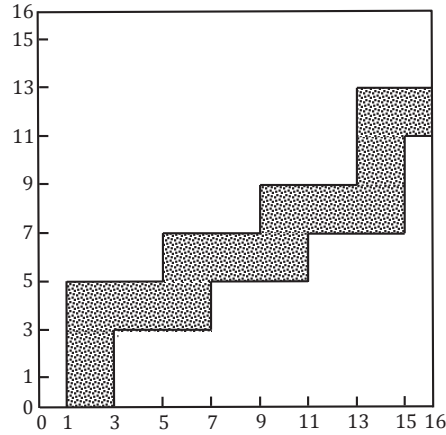


FIG. 4. Sawtooth coefficient jump domain.

The next set of experiments examines 3D diffusion operators. We examine the problem

$$(5.2) \quad u_{xx} + \epsilon_y u_{yy} + \epsilon_z u_{zz} = f$$

using classic second-order seven-point finite differencing approximations on regular

TABLE 2

2D non-Galerkin results for various γ , sawtooth coefficient jump problem, AMG preconditioning GMRES.

γ	Refinement	ρ_{AMG}	Op. comp.	Work	Max stencil
1.0	0	0.25	2.16	14	11
	1	0.30	2.18	17	16
	2	0.51	2.19	30	14
	3	0.67	2.19	50	15
0.03	0	0.14	2.18	10	12
	1	0.15	2.19	11	14
	2	0.20	2.20	12	15
	3	0.25	2.20	15	20
0.01	0	0.14	2.19	10	13
	1	0.16	2.19	11	15
	2	0.20	2.20	12	17
	3	0.26	2.20	15	20
0.0	0	0.15	2.20	11	16
	1	0.17	2.20	11	17
	2	0.20	2.21	13	23
	3	0.28	2.21	16	23

grids and also linear tetrahedral discretizations on unstructured isotropic meshes. The PDE coefficients considered are $\epsilon_y = 1.0$, $\epsilon_z = 0.001$ and $\epsilon_y = 1.0$, $\epsilon_z = 1.0$.

The results are broadly similar, so we give a representative set in Table 3 for the linear tetrahedral example and isotropic coefficients ($\epsilon_y = 1.0$, $\epsilon_z = 1.0$). The non-Galerkin approach is robust, preserves the Galerkin AMG convergence rate, and yields a stencil size reduction of 50%. One important note is that the operator complexity reduction is not always as dramatic as depicted in Table 3. For instance, for the seven-point finite-differencing Poisson operator, the operator complexity reduction is only from 2.9 to 2.8, but the non-Galerkin approach still enjoys a roughly 50% stencil size reduction.

TABLE 3

3D non-Galerkin results for two γ choices, isotropic diffusion, AMG preconditioning GMRES.

γ	Refinement	ρ_{AMG}	Op. comp.	Work	Max stencil
1.0	0	0.18	1.4	8	45
	1	0.26	1.5	10	72
	2	0.32	1.7	14	77
	3	0.40	1.8	18	78
0.03	0	0.18	1.5	8	52
	1	0.24	1.6	10	76
	2	0.30	1.8	14	81
	3	0.36	1.9	17	81
0.0	0	0.18	1.6	8	73
	1	0.24	1.8	11	91
	2	0.30	2.2	17	150
	3	0.38	2.6	24	174

5.2. Elasticity results. We next examine the proposed approach for isotropic linearized elasticity problems, where $B \in \mathbb{R}^{n,6}$ and is equal to the six rigid-body

modes. Linearized elasticity is defined by

$$(5.3) \quad -\operatorname{div} \left(\lambda \operatorname{tr} \left((\nabla \mathbf{u} + \nabla \mathbf{u}^T) / 2 \right) I + \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \right) = f,$$

where λ and μ are the Lamé parameters, I is the identity matrix, and $\operatorname{tr}()$ is the trace function. The GetFem++ package [13] is used to discretize. The test problem examined is a steel tripod corresponding to the mesh in Figure 5. Here, the tripod is discretized using linear tetrahedral elements, and a downward external force is applied to the top of the tripod.

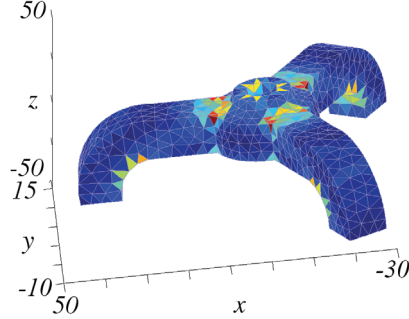


FIG. 5. Example tetrahedral tripod mesh.

We experiment with a different AMG solver than in section 5.1 because the interpolation operators for RS-AMG capture only the constant vector. Hence, the experiments with RS-AMG show no performance difference when comparing Algorithm 3 with Algorithm 4, and B equals the six rigid-body modes. Essentially, if $P^T A P$ does not accurately represent the rigid-body modes (as in the case of RS-AMG), then the stencil collapsing approach does not need to account for them. So, we consider instead a smoothed aggregation (SA) approach [12] that allows $\operatorname{span}(P)$ to capture all six of the rigid-body modes. Hence, $P^T A P$ accurately represents all six of the rigid-body modes, thus making it important for the stencil collapsing approach to account for all six modes. We also choose an SA approach to show the generality of the proposed non-Galerkin method.

While SA generally does not associate C -points with the coarse grid, this set of unknowns is readily available from many standard aggregation algorithms [19] where the aggregate seed points can be used to form the set of C -points. This is done explicitly in [12], where the set of C -points is used to form P_I . The other parameters for the SA method [12] are as follows. V(2,2) cycles of weighted-Jacobi are again used to accelerate GMRES with a relative stopping criterion of 10^{-8} and a restart value of 15. The Jacobi weight is $1/\rho(D^{-1}A)$, where D is the diagonal of A . For consistency, the strength of connection method is the classical Ruge–Stüben method with a tolerance of 0.35. Standard greedy aggregation [19] is used.

The convergence results for the tripod example are given in Table 4. Three consecutive refinements are considered for increasing matrix sizes of roughly 3 000, 17 000, and 109 000. The second column denotes the chosen values for B , with “RBM” standing for the six rigid-body modes, and $B = \mathbf{1}$ representing the standard stencil collapsing of Algorithm 3. The choice of $B = \mathbf{1}$ is similar to collapsing with the three displacement rigid-body modes, which likely explains why Algorithm 3 performs well.

The proposed approach matches the convergence rate of the Galerkin approach for $\gamma = 0.03$, with a significant drop in maximum stencil size. For smaller values

of γ , such as $\gamma = 0.03$, the results are nearly identical for B equalling the six rigid-body modes or the constant. This is not surprising because classical multigrid schemes based on $B = \mathbf{1}$ are known to be effective for some simpler elasticity problems, such as this example. The use of the six rigid-body modes becomes important when the non-Galerkin algorithm is stressed with the very restricted sparsity pattern of $\gamma = 1.0$. Here, collapsing with respect to the rigid-body modes, but not $B = \mathbf{1}$, allows for regaining the original work per digit of accuracy numbers of the Galerkin approach.

TABLE 4

Elasticity non-Galerkin results for various γ and collapsing strategies, AMG preconditioning GMRES.

γ	B	Refinement	ρ_{AMG}	Op. comp.	Work	Max stencil
1.0	$\mathbf{1}$	0	0.45	1.2	12	42
		1	0.50	1.2	16	72
		2	0.64	1.2	24	97
1.0	RBM	0	0.44	1.2	12	42
		1	0.48	1.2	14	72
		2	0.54	1.2	18	97
0.03	$\mathbf{1}$ or RBM	0	0.40	1.3	11	56
		1	0.44	1.4	15	138
		2	0.48	1.5	18	245
0.0	$\mathbf{1}$ or RBM	0	0.39	1.4	11	69
		1	0.43	1.5	16	182
		2	0.47	1.6	20	346

5.3. Parallel results. In this section, we present parallel results for the simple scalar seven-point 3D isotropic diffusion finite-difference operator with Dirichlet boundary conditions. The goal is to demonstrate the potential of the algorithm to speed up parallel AMG solve phase times. We therefore choose the simple seven-point model problem that nonetheless yields an unscalable growth in stencil size. The parallel stencil collapsing is identical to the serial algorithm, with a few exceptions. The collapsing is done with a binary strength-of-connection matrix (i.e., each strong connection has a value of 1.0, and each weak connection has a value of 0.0). This is done because binary strength-of-connection matrices are what is currently available in *hypr*. Implementing the complete stencil collapsing approach from Algorithm 3 using standard strength matrices is left for future work.

The parallel studies carried out are weak scaling tests with 25,000 unknowns per core on two different machines, one with a fast InfiniBand QDR interconnect and 6144 cores and the other with a slow InfiniBand DDR interconnect and 4096 cores. Thus, the largest problem sizes considered are 153.6 and 102.4 million unknowns, respectively. V(1,1) cycles using hybrid Gauss–Seidel [3] are used to precondition GMRES to within a relative residual tolerance of 10^{-8} using a restart parameter of 10. We note that the non-Galerkin algorithm has also been run in symmetrized mode with CG and the results have not qualitatively differed from the results below.

Two different AMG scenarios are explored in parallel; one uses a “best practices” set of parameters, which the *hypr* developers commonly recommend, and the other uses classical parallel AMG parameters. The *hypr* parameters for best practices are hybrid symmetric Gauss–Seidel (relax type 6), HMIS [16] coarsening (coarsen type 10), extended classical modified interpolation [15] (interpolation type 6), aggressive coarsening [16] on the finest level (aggressive num levels 1), truncated interpolation to

five entries per row (PMaxElmts option equals 5), and a strength-of-connection tolerance of 0.25. The *hypr* parameters for classical parallel AMG are hybrid symmetric Gauss–Seidel (relax type 6), classical modified interpolation (interpolation type 0), Falgout coarsening [9] (coarsen type 0), and a strength-of-connection tolerance of 0.55. We note that it is typical to use a higher strength tolerance in this case. We examine the non-Galerkin approach for $\gamma = 0.03$, which is a representative γ value that worked well over all the serial test problems, and for $\gamma = 0.0$, which corresponds to standard Galerkin AMG.

We first report in Tables 5 and 6 the convergence, operator complexity, work, and maximum stencil size over all levels for the best practices parameters and classical parallel AMG parameters, respectively. We do this for the four largest core counts (problem size equals 25K times the core count). The results are similar to the serial case in that the non-Galerkin approach attains a similar AMG convergence rate but results in a significantly smaller maximum stencil size. For the case of classical parallel AMG, which has always been plagued by high complexities, there is also a significant reduction in operator complexity that makes this method more practical. In addition, the convergence rate for non-Galerkin is noticeably better for classical parallel AMG, although the reasons for this are difficult to discern in the multilevel setting. In the two-level setting, the convergence rates do not significantly differ.

TABLE 5

“Best practices” AMG results for two γ choices, isotropic diffusion, AMG preconditioning GMRES.

γ	ncores	ρ_{AMG}	Op. comp.	Work	Max stencil
0.03	768	0.33	1.25	10	82
	1536	0.33	1.26	10	81
	3072	0.36	1.26	11	85
	6144	0.37	1.25	11	96
0.0	768	0.33	1.37	11	184
	1536	0.36	1.37	12	188
	3072	0.40	1.37	14	191
	6144	0.39	1.37	13	206

TABLE 6

“Classical parallel AMG” results for two γ choices, isotropic diffusion, AMG preconditioning GMRES.

γ	ncores	ρ_{AMG}	Op. comp.	Work	Max stencil
0.03	768	0.19	4.8	27	381
	1536	0.20	5.0	29	380
	3072	0.20	5.2	30	435
	6144	0.21	5.5	32	463
0.0	768	0.22	8.4	51	1266
	1536	0.24	9.0	58	1286
	3072	0.27	9.4	66	1405
	6144	0.29	9.7	72	1479

Next, we report speedup and time to solution for the AMG solve phase in order to verify that the observed reduction in stencil size results in a faster runtime. Since this is a feasibility study and in order to simplify the parallel implementation, the non-Galerkin part of the AMG setup phase has not been written with attention to

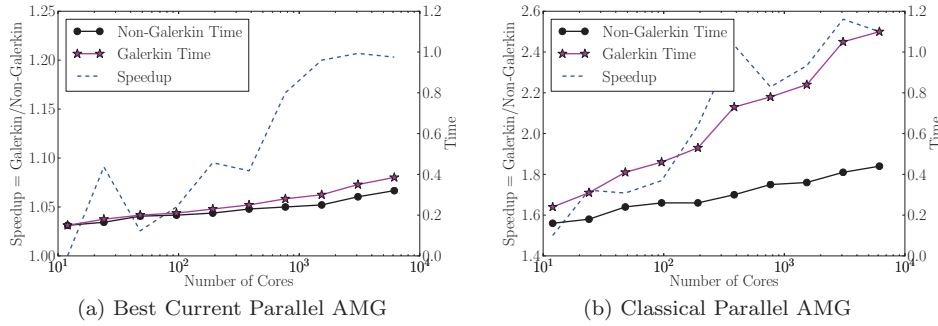


FIG. 6. *Speedup and AMG solve phase timings comparing non-Galerkin AMG and Galerkin AMG on machine with fast network.*

performance. So, we do not report setup phase timings, as they would not be informative. We do note that even with the unoptimized code, we always see a reduction in setup phase time for classical parallel AMG, and for best practices AMG, some machines also already show a reduction in setup phase time. A full timing comparison is ongoing work and will require optimizing the setup phase and parameters such as different γ choices on coarse levels.

Figures 6a and 6b show the speedup and timings for the best practices scenario and classical parallel AMG scenario, respectively, on the machine with a fast interconnect. The speedup is on the left axis for the dotted line, and the time to solution for the AMG solve phase is on the right axis for the two solid lines depicting the Galerkin and non-Galerkin approaches. For the best practices scenario, there is a speedup of 15–20%, but for the classical parallel AMG scenario, the speedup grows to 250%. Figures 7a and 7b depict similar results for the machine with a slow interconnect, but with more pronounced speedups of 150% and 400% for the best practices and classical parallel AMG scenarios, respectively.

These results are encouraging, in part because the speedup grows with number of cores. Thus, we expect this approach to yield even better speedups when moving to current Petascale machines (10^6 cores) and also to next-generation exascale machines (10^6 – 10^9 cores).

Moreover, the results raise the question of whether the current best practices in AMG can be avoided altogether in favor of a more classical AMG approach that leverages non-Galerkin coarse grids. For instance, if the overall time is compared between the best practices and classical parallel AMG scenarios for each machine, it is observed that the non-Galerkin approach for the classical scenario is competitive with the Galerkin approach for the best practices scenario. This rethinking of parallel AMG is attractive because the current best practices for reducing stencil size and communication (i.e., aggressive coarsening and extended interpolation) also worsen the AMG convergence rate, whereas non-Galerkin coarse grids target the same reduced stencil size and associated reduced communication, but with no deterioration in the AMG convergence rate. For example, our parallel tests for a 27-point finite element discretization of the Poisson problem already show the fastest AMG solve phase time is for classical parallel AMG coupled with the non-Galerkin approach. Regardless of which AMG scenario is used, the non-Galerkin approach provides a parallel performance benefit.

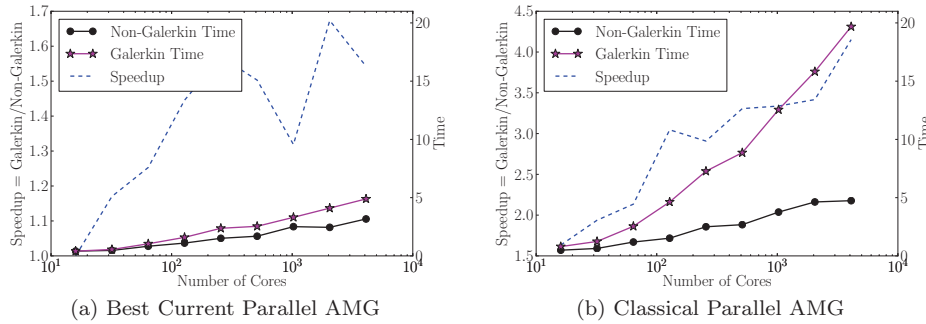


FIG. 7. Speedup and AMG solve phase timings comparing non-Galerkin AMG and Galerkin AMG on machine with slow network.

6. Comments/conclusions. In conclusion, the proposed non-Galerkin approach is an effective method at reducing coarse grid stencil size and parallel communication, while having a negligible effect on the AMG convergence rate for the problems considered here. In particular, the method relies primarily on row-wise computations in parallel, thus making it a relatively inexpensive method with respect to communication. Additionally, as indicated by our experiments, the proposed method is applicable to most AMG codes, including classical RS-AMG and SA.

There are some outstanding issues, including the importance of maintaining symmetry and definiteness on coarse grids. While theoretically attractive, we have not observed maintenance of symmetry being critical to performance, even up to 150 million unknowns in parallel. However, we will maintain this option in our code as we continue experiments on more real-world test problems. Maintaining definiteness on coarse grids, outside of the case of M-matrices, will continue to be a future research issue.

REFERENCES

- [1] R. E. ALCOUFFE, A. BRANDT, J. E. DENDY, JR., AND J. W. PAINTER, *The multi-grid method for the diffusion equation with strongly discontinuous coefficients*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 430–454.
- [2] S. F. ASHBY AND R. D. FALGOUT, *A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations*, Nuclear Sci. Engrg., 124 (1996), pp. 145–159. UCRL-JC-122359.
- [3] A. H. BAKER, R. D. FALGOUT, Tz. V. KOLEV, AND U. M. YANG, *Multigrid smoothers for ultraparallel computing*, SIAM J. Sci. Comput., 33 (2011), pp. 2864–2887.
- [4] W. N. BELL, L. N. OLSON, AND J. SCHRODER, *PyAMG: Algebraic multigrid solvers in Python v2.0*, <http://www.pyamg.org>, 2011.
- [5] A. BRANDT, *General highly accurate algebraic coarsening*, Electron. Trans. Numer. Anal., 10 (2000), pp. 1–20.
- [6] A. BRANDT, S. F. MCCORMICK, AND J. W. RUGE, *Algebraic multigrid (AMG) for sparse matrix equations*, in Sparsity and Its Applications, D. J. Evans, ed., Cambridge University Press, Cambridge, UK, 1984, pp. 257–284.
- [7] W. L. BRIGGS, V. E. HENSON, AND S. F. MCCORMICK, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, 2000.
- [8] H. GAHVARI, A. H. BAKER, M. SCHULZ, U. M. YANG, K. E. JORDAN, AND W. GROPP, *Modeling the performance of an algebraic multigrid cycle on HPC platforms*, in Proceedings of the International Conference on Supercomputing, ICS '11, ACM, New York, 2011, pp. 172–181.

- [9] V. HENSON AND U. YANG, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177.
- [10] *hypr: High performance preconditioners*. <http://www.llnl.gov/CASC/hypr/>.
- [11] Y. NOTAY, *Convergence analysis of perturbed two-grid and multigrid methods*, SIAM J. Numer. Anal., 45 (2007), pp. 1035–1044.
- [12] L. N. OLSON, J. B. SCHRODER, AND R. S. TUMINARO, *A general interpolation strategy for algebraic multigrid using energy minimization*, SIAM J. Sci. Comput., 33 (2011), pp. 966–991.
- [13] Y. RENARD, *The GetFem++ Project*, http://download.gna.org/getfem/doc/getfem_project.pdf.
- [14] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid (AMG)*, in Multigrid Methods, S. F. McCormick, ed., Frontiers Appl. Math. 3, SIAM, Philadelphia, 1987, pp. 73–130.
- [15] H. DE STERCK, R. D. FALGOUT, J. W. NOLTING, AND U. M. YANG, *Distance-two interpolation for parallel algebraic multigrid*, Numer. Linear Algebra Appl., 15 (2008), pp. 115–139. UCRL-JRNL-230844.
- [16] H. DE STERCK, U. M. YANG, AND J. J. HEYS, *Reducing complexity in parallel algebraic multigrid preconditioners*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 1019–1039.
- [17] E. TREISTER, *Sparsified coarsening multigrid for convection diffusion*, Student paper at the Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, 2013.
- [18] U. TROTTEMBERG, C. OOSTERLEE, AND A. SCHÜLLER, *Multigrid*, Academic Press, London, 2001.
- [19] P. VANĚK, J. MANDEL, AND M. BREZINA, *Algebraic multigrid based on smoothed aggregation for second and fourth order problems*, Computing, 56 (1996), pp. 179–196.
- [20] R. WIENANDS AND I. YAVNEH, *Collocation coarse approximation in multigrid*, SIAM J. Sci. Comput., 31 (2009), pp. 3643–3660.