SIAM J. SCI. COMPUT.
Vol. 39, No. 5, pp. S723–S756

A ROOT-NODE–BASED ALGEBRAIC MULTIGRID METHOD*

THOMAS A. MANTEUFFEL†, LUKE N. OLSON‡, JACOB B. SCHRODER§, AND BEN S. SOUTHWORTH†

Abstract. This paper provides a unified and detailed presentation of root-node–style algebraic multigrid (AMG). AMG is a popular and effective iterative method for solving large, sparse linear systems that arise from discretizing partial differential equations. However, while AMG is designed for symmetric positive definite (SPD) matrices, certain SPD problems, such as anisotropic diffusion, are still not adequately addressed by existing methods. Non-SPD problems pose an even greater challenge, and in practice AMG is often not considered as a solver for such problems. The focus of this paper is on so-called root-node AMG, which can be viewed as a combination of classical and aggregation-based multigrid. An algorithm for root-node AMG is outlined, and a filtering strategy is developed, which is able to control the cost of using root-node AMG, particularly on difficult problems. New theoretical motivation is provided for root-node and energy-minimization as applied to symmetric as well nonsymmetric systems. Numerical results are then presented demonstrating the robust ability of root-node AMG to solve nonsymmetric problems, systems-based problems, and difficult SPD problems, including strongly anisotropic diffusion, convection-diffusion, and upwind steady-state transport, in a scalable manner. New detailed estimates of the computational cost of the setup and solve phases are given for each example, providing additional support for root-node AMG over alternative methods.

Key words. multigrid, algebraic multigrid, root-node, energy minimization, interpolation smoothing, anisotropic diffusion

AMS subject classifications. 65F10, 65M22, 65M55

DOI. 10.1137/16M1082706

1. Introduction.

Algebraic multigrid (AMG) methods such as classical AMG (CF AMG[1]) [11, 42] and smoothed aggregation AMG (SA AMG) [48] are efficient solution techniques for large, sparse linear systems. AMG was developed specifically for symmetric positive-definite (SPD) systems that arise from the discretization of elliptic partial differential equations (PDEs), and software packages such as BoomerAMG [28] in the hypre library [26] demonstrate its parallel scalability to hundreds of thousands of cores [3].

AMG targets solution of a sparse linear system (typically SPD),

$$A\mathbf{x} = \mathbf{b}, \tag{1}$$

*Received by the editors July 5, 2016; accepted for publication (in revised form) May 22, 2017; published electronically October 26, 2017.

http://www.siam.org/journals/sisc/39-5/M108270.html

Funding: This research was conducted with government support and awarded by the DoD, Air Force Office of Scientific Research, and National Defense Science and Engineering Graduate (NDSEG) Fellowship 32 CFR 168a. Portions of this work were sponsored by the Air Force Office of Scientific Research under grant FA9550–12–1–0478. This work was also performed under the auspices of the U.S. Department of Energy under grants (SC) DE-FC02–03ER25574 and (NNSA) DE-NA0002376 and by Lawrence Livermore National Laboratory under contracts B600360, DE-AC52–07NA27344, and LLNL-JRNL-695797.

†Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309 (tmanteuff@colorado.edu, ben.southworth@colorado.edu).

‡Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (lukeo@illinois.edu, http://lukeo.cs.illinois.edu).

§Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551 (schroder2@llnl.gov, http://people.llnl.gov/schroder2).

[1]Classical AMG, or so-called Ruge–Stüben AMG, uses a splitting of the degrees-of-freedom (DOFs) into coarse *C-points* and fine *F-points*, leading to the abbreviation CF.

with $O(n)$ work, where $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$. This optimality is achieved through two complementary parts of a multigrid method, relaxation and coarse-grid correction, which together uniformly damp error of all frequencies (see section 2).

However, there are certain SPD systems and many nonsymmetric systems for which AMG continues to struggle. Problems with strongly anisotropic components, and problems arising in particle transport, advective flow calculations, and strongly varying material properties, among others, challenge the standard approaches to AMG, thus highlighting the need for more robust methods. There have been a number of efforts in recent years to improve the convergence and scope of applicability of AMG. Adaptive methods focus on improving the multigrid hierarchy through trial cycles in the setup phase [16, 17, 18, 22]. Other methods focus on modified or improved strength measures when forming coarse grids [7, 9, 10, 12, 13, 15, 22, 31, 36, 39]. Furthermore, generalizing interpolation through energy minimization [41] and other methods [20, 23, 27, 50] is used to improve the accuracy of interpolation between grid levels. Nevertheless, many problems remain difficult for AMG to solve, while many "robust" AMG methods suffer from high computational cost.

An AMG solver consists of a hierarchy of matrices, $\{A_\ell\}$, with the initial matrix on level $\ell = 0$, $A_0 := A$, and progressively smaller matrices on levels $\ell = 1, 2, \ldots, L$. Interpolation and restriction operators, also known as transfer operators, transfer vectors between different levels of the hierarchy. For a given matrix A_ℓ, the next "coarser" matrix in the hierarchy (level $\ell + 1$) is generally developed in one of two different ways: using a CF-splitting of points (CF AMG) or using an aggregation of points (SA AMG). A CF-splitting splits the set of all DOFs of matrix A_ℓ into a coarse set of C-points and a fine set of F-points, with C-points corresponding to DOFs on the coarse grid. Transfer operators are then defined using the CF-splitting, where values at C-points are restricted and interpolated by injection, and values at F-points use a linear combination of connected neighboring points. In SA AMG, a measure of the strength-of-connection (SOC) between nodes is used to form "aggregates," which are disjoint sets of strongly connected nodes, where each aggregate represents one node on the coarse grid, and transfer operators are formed based on aggregates. A more detailed look at SA AMG and CF AMG is given in section 2.

Root-node AMG (RN AMG) uses a hybrid approach, wherein SA-type SOC and aggregation routines are used to form aggregates. In each aggregate, one node is chosen to be the "root-node," which corresponds to a C-point, and other nodes are designated as F-points. Transfer operators are then formed based on this CF-splitting together with aggregation.

RN AMG was initially identified in [41] as a small part of a general, energy-minimizing framework for forming interpolation operators in AMG. The work in [45] implemented the RN approach and demonstrated its potential as an effective and scalable solver for strongly anisotropic, non–grid-aligned diffusion operators—problems which have proven difficult for other multigrid methods. However, certain problems required a large computational cost, especially in the setup of the method.

This paper provides, for the first time, a unified and detailed presentation of RN AMG and how it combines many of the benefits of CF AMG and SA AMG. RN AMG allows for classical pointwise decisions in the setup, to help control complexity[2] and provide theoretical motivation, as well as aggregation-style construction, to facilitate the design of a multigrid solver based on the spectral behavior of the problem.

[2] "Complexity" or "cost" refers to the overall computational cost of the method in terms of floating point operations.

New theoretical motivation is given for the pairing of RN AMG with energy minimization, in both the symmetric and nonsymmetric cases, proving the equivalence of the energy-minimization process to minimizing the difference with an optimal form of interpolation. Moreover, a new interpolation filtering strategy is developed to limit the complexity of the method, which proves critical for problems that require large sparsity patterns in transfer operators. Last, a numerical survey is presented to highlight the robustness and flexibility of RN AMG in comparison to CF AMG and SA AMG, including scalable convergence for strongly anisotropic diffusion problems and a discontinuous, upwind discretization of the steady-state transport equation. For each test problem, a detailed measure of computational cost or complexity is provided, a novel addition to AMG literature, which provides a complete picture of a method when coupled with convergence factors.

In section 2, background information on AMG methods is discussed, including current limitations and the basic motivation for an RN-type algorithm. The RN AMG method and algorithm are presented in section 3, along with a discussion of computational complexity and a filtering process proposed to address cases of high complexity. Section 4 provides theoretical motivation for RN AMG. Numerical results are provided in section 5, and conclusions and future work discussed in section 6.

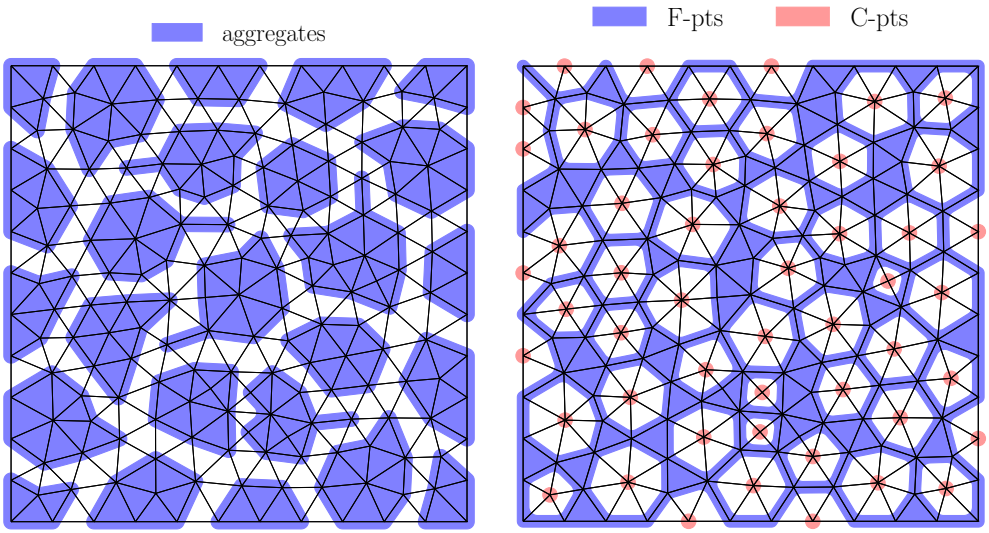
2. Background. Multigrid methods, such as SA AMG and CF AMG, involve two phases: (i) the setup phase, where a multilevel solver hierarchy is constructed, and (ii) the solve phase, where the constructed solver hierarchy is applied to solving the linear system (1) to a desired tolerance. SA AMG and CF AMG are distinguished by the setup phase; that is, once a hierarchy is constructed, both methods execute the solve phase in the same fashion.

In the following, a multigrid hierarchy consists of a set of matrices starting with an initial, fine matrix $A_0 \equiv A \in \mathbb{R}^{n\times n}$. Matrices for additional levels in the hierarchy, $A_\ell \in \mathbb{R}^{n_\ell \times n_\ell}$, are then constructed based on interpolation and restriction operators, $P_\ell : \mathbb{R}^{n_{\ell+1}} \to \mathbb{R}^{n_\ell}$ and $R_\ell : \mathbb{R}^{n_\ell} \to \mathbb{R}^{n_\ell+1}$, respectively, via $A_{\ell+1} = R_\ell A_\ell P_\ell$, where $n_{\ell+1} < n_\ell$. In the case of an SPD matrix, restriction $R = P^T$.

The AMG solve phase iterates using two complementary parts: relaxation—e.g., weighted Jacobi—to reduce the high-energy error that is associated with large eigenvalues in the operator, and coarse-grid correction, which targets algebraically smooth error, $A_0\mathbf{e} \approx 0$, associated with the small eigenvalues. A two-grid solve proceeds as follows. Prerelaxation on $A_0\mathbf{x}_0 = \mathbf{b}_0$ is applied, and the resulting residual is then restricted to the coarse grid, $\mathbf{b}_1 = R_0(\mathbf{b}_0 - A_0\mathbf{x}_0)$, which serves as the right-hand side for the coarse-grid equation, $A_1\mathbf{x}_1 = \mathbf{b}_1$, where $A_1 = R_0A_0P_0$. The solution, $\mathbf{x}_1$, provides a coarse-grid error correction that is interpolated back to the fine grid, $\mathbf{x}_0 \leftarrow \mathbf{x}_0 + P_0\mathbf{x}_1$. Last, postrelaxation is applied to the updated $\mathbf{x}_0$. Together, these three steps form a two-level multigrid V(ν_{pre}, ν_{post})-cycle, where ν_{pre} refers to the number of prerelaxations, and ν_{post} to the number of postrelaxations. A full solve then consists of using successive V-cycles to iterate on a vector until the relative residual norm, $\|\mathbf{r}_0\|/\|\mathbf{b}_0\|$, is less than some tolerance—e.g., 10^{-8}.

The effectiveness of this complementary process is explained by considering the corresponding two-grid error propagation operator. Let $\mathbf{e}^{(0)}$ be the initial error in approximating the solution to (1), and let $\mathbf{e}^{(0)} \leftarrow G\mathbf{e}^{(0)}$ represent the error propagator for the relaxation method—e.g., $G = I - \omega D^{-1}A$ in the case of weighted Jacobi. Then (dropping subscripts), the error after a two-grid cycle is given by

$$\mathbf{e}^{(1)} \leftarrow G^{\nu_2}\left(I - P(RAP)^{-1}RA\right)G^{\nu_1}\mathbf{e}^{(0)}, \tag{2}$$



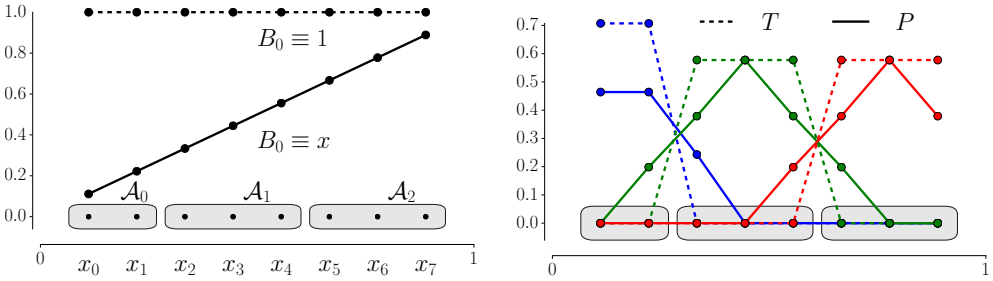
(a) Aggregation. (b) C/F splitting.

FIG. 1. *Example SA AMG and CF AMG coarsening for a linear finite element approximation to a Laplace operator. The fine level problem has* 191 *DOFs. In this example, aggregation yields* 25 *coarse DOFs (aggregates), while a CF-splitting yields* 51 *coarse DOFs (C-points).*

where $P(RAP)^{-1}RA$ is a projection onto $\mathcal{R}(P)$. In the case of SPD A, $R = P^T$, and this is an A-orthogonal projection. In either case, if $Ge^{(0)} \in \mathcal{R}(P)$, then the iteration is exact. In other words, if interpolation is complementary and accurate for modes not effectively reduced by relaxation, then error reduction with (2) will be large.

AMG methods attempt to automatically determine interpolation and coarse-grid operators (P_ℓ and A_ℓ) that yield optimal error reduction with (2). The two standard approaches are SA AMG and CF AMG, which are outlined in sections 2.1 and 2.2, respectively. Broadly, SA AMG defines a coarse problem through an *aggregation* of nodes (see Figure 1(a), while CF AMG defines a coarse problem through a splitting of the DOFs into coarse C-points and fine F-points (see Figure 1(b)). Each offers advantages, as noted in the following descriptions.

2.1. AMG based on smoothed aggregation. The effectiveness of SA AMG relies on a priori knowledge of algebraically smooth error in the form of candidate vectors, B. These vectors generally represent the lowest energy modes of the governing PDE with no boundary conditions—e.g., the constant for diffusion and rigid-body-modes for elasticity [48]. Using B (possibly determined a priori), the setup phase first uses a strength measure on the connectivity of nodes to define an SOC matrix, S, which is used to identify so-called aggregates (see Figure 1(a)). The goal of the strength measure is to ensure that algebraically smooth error at each DOF in an aggregate strongly correlates with algebraically smooth error at other DOFs in that aggregate. This holds for model problems, where algebraically smooth error over each aggregate is well represented by the restriction of the candidate vectors to the aggregate. Consequently, this *injection* of the candidate vectors over each aggregate is what leads to the initial representation of interpolation, termed the tentative interpolation operator T, with candidate vectors exactly in the range of T. Each aggregate



(a) Three aggregates and two candidate vectors.

(b) Columns of interpolation using a single candidate with SA AMG.

FIG. 2. *1D Laplace example using SA AMG.*

corresponds to one block in the block-diagonal operator T, with one column for each candidate and one row for each point in the aggregate.

As an example, consider the one-dimensional (1D) Laplace operator on an eight-node mesh, using standard finite differences, with candidate vectors $B = [\mathbf{1}, \mathbf{x}]$. This yields three aggregates, $\mathcal{A}_i$, $i = 0, 1, 2$, as shown in Figure 2(a). The aggregation pattern matrix, C, coupled with the candidate vectors, B, yield the following:

(3)
$$C = \begin{bmatrix} * & & \\ * & & \\ & * & \\ & * & \\ & * & \\ & & * \\ & & * \\ & & * \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1/9 \\ 1 & 2/9 \\ 1 & 3/9 \\ 1 & 4/9 \\ 1 & 5/9 \\ 1 & 6/9 \\ 1 & 7/9 \\ 1 & 8/9 \end{bmatrix} \rightarrow T = \begin{bmatrix} 1 & -1 & & & & \\ 1 & 1 & & & & \\ & & 1 & -1 & & \\ & & 1 & 0 & & \\ & & 1 & 1 & & \\ & & & & 1 & -1 \\ & & & & 1 & 0 \\ & & & & 1 & 1 \end{bmatrix} D,$$

where $D = \mathrm{diag}([2, 2, 3, 2, 3, 2])^{-1/2}$ is a diagonal matrix to normalize each column in the l^2-norm. Each block of T in (3) is an orthogonal basis for the restriction of B to each aggregate.

In the case of a single candidate vector, $B = [\mathbf{1}]$, T consists of columns 0, 2, and 4 in (3). These columns are plotted in Figure 2(b) (dashed lines). It is important to note that each column is nonzero only on its respective aggregate. To improve the accuracy of interpolation for algebraically smooth modes—i.e., to make $\mathcal{R}(P)$ more complementary to relaxation—the columns of the tentative interpolation operator are smoothed—e.g., with weighted Jacobi—to form P (Figure 2(b), solid lines). As the range of P becomes richer, so does the nonzero footprint in the operator. Indeed, nonzero elements of the middle column of P in Figure 2(b) (solid green) now overlap into the neighboring aggregates.

Given a set of candidate vectors B on the current grid that are exactly represented in the range of T, coarse-grid candidate vectors are constructed as the preimage of B under T. The motivation is that the preimage of low-energy vectors for A_ℓ should be low-energy vectors for $A_{\ell+1}$. Let the orthogonal projection onto $\mathcal{R}(T)$ be given by $\pi_T = T(T^T T)^{-1} T^T$. Since the columns of T are orthonormal, $\pi_T = TT^T$, and requiring $\pi_T B = B$, that is $B \in \mathcal{R}(T)$, results in a coarse-grid preimage of B under T given by $B_c = T^T B$. Since B is assumed to be a low energy mode, forming P by

applying smoothing iterations to T to improve multilevel convergence [46] keeps B close to the range of P.

A general SA AMG setup algorithm is given in Algorithm 1.

Algorithm 1: `SA_setup()`

Input: A_0: fine-grid operator
B_0: fine-grid candidate vectors
`max_size`: threshold for maximum size of coarsest problem

Output: $A_1, \ldots, A_L$, $P_0, \ldots, P_{L-1}$

1 $\ell = 0$
2 **while** `size`$(A_\ell) >$ `max_size`
3 $S_\ell =$ `strength`(A_ℓ) {strength-of-connection}
4 $\mathcal{A}_\ell =$ `aggregate`(S_ℓ) {aggregation}
5 T_ℓ, $B_{\ell+1} =$ `inject`$(\mathcal{A}_\ell, B_\ell)$ {form tentative interpolation and coarse candidates}
6 $P_\ell =$ `smooth`(A_ℓ, T_ℓ) {smooth T_ℓ}
7 $A_{\ell+1} = P_\ell^T A_\ell P_\ell$ {coarse-grid operator}
8 $\ell = \ell + 1$

2.2. AMG based on coarse-fine splittings. In contrast to SA AMG, CF AMG builds a multilevel hierarchy through a CF-splitting. On each level, $\ell = 0, 1, \ldots, L$, the index set of DOFs, $\Omega^\ell = \{0, \ldots, n_\ell\}$, is split into $\Omega^\ell = \mathcal{C}_\ell \cup \mathcal{F}_\ell$, where $\mathcal{C}_\ell \cap \mathcal{F}_\ell = \emptyset$. The set $\mathcal{C}_\ell$ defines the coarse-level DOFs so that $n_{\ell+1} = |\mathcal{C}_\ell|$. Similar to SA AMG, a strength measure, S, is used to determine the splitting so that algebraically smooth errors at F-points are well approximated by evaluating neighboring C-points. Then interpolation, $P : \mathbb{R}^{n_{\ell+1}} \to \mathbb{R}^{n_\ell}$, is formed as

$$P = \begin{bmatrix} W \\ I \end{bmatrix} \begin{matrix} \} \text{ F-points,} \\ \} \text{ C-points,} \end{matrix} \tag{4}$$

where $W \in \mathbb{R}^{|\mathcal{F}_\ell|} \times \mathbb{R}^{|\mathcal{C}_\ell|}$ is a sparse matrix with entries chosen to approximate smooth error at F-points (rows of W) as a linear combination of C-points (columns of W).

The form of interpolation in (4) highlights two attributes of interpolation in CF AMG that distinguishes it from SA AMG. The first is that the source of complexity and accuracy is clear: the number of nonzeros in W controls both the accuracy and cost of interpolation. This is explored further in section 2.4. Second, coarse values are *injected* to the finer grid through the identity in the bottom block. This in turn ensures linear independence of the columns of P, an important feature not guaranteed in SA AMG.

A general CF AMG setup algorithm is given in Algorithm 2.

2.3. Benefits and limitations. In CF AMG, it is assumed that the constant vector is representative of the near null space of the underlying problem, as in the case of a Poisson problem. As a result, convergence can degrade if this is not an accurate assumption. Moreover, the interpolation formulas for CF AMG are static and offer no immediate ways to improve them for more difficult problems.

One benefit of CF AMG is that it provides a structure for controlling sparsity through the determination of weights (see line 5 in Algorithm 2). Each row of W

Algorithm 2: `CF_setup()`

Input: A_0: fine-grid operator
`max_size`: threshold for max size of coarsest problem

Output: $A_1, \ldots, A_L$,
$P_0, \ldots, P_{L-1}$

$\ell = 0$
while `size`$(A_\ell) >$ `max_size`
  $S_\ell =$ `strength`(A_ℓ) {strength-of-connection}
  $\mathcal{C}_\ell, \mathcal{F}_\ell =$ `splitting`(S_ℓ) {C/F-splitting}
  $W =$ `weights`$(S_\ell, A_\ell, \mathcal{C}_\ell, \mathcal{F}_\ell)$ {interpolation weights}
  $P_\ell = \begin{bmatrix} W \\ I \end{bmatrix}$ {form interpolation}
  $A_{\ell+1} = P_\ell^T A_\ell P_\ell$ {coarse-grid operator}
  $\ell = \ell + 1$

(see (4)) represents an interpolation formula for a given F-point from surrounding C-points. As a result, this leads to simple filtering strategies in P [23] that eliminate small entries in each row. In SA AMG, interpolation is based on having algebraically smooth columns of P. Consequently, existing methods to explicitly filter entries in P are limited, as removing entries can greatly reduce smoothness [20]. The structure of interpolation operators in CF AMG also allows for theoretical results that are not feasible for arbitrary transfer operators, as in SA AMG [25, 38, 49], and which are used in section 4 to motivate RN AMG.

For problems in which a single global vector adequately represents the algebraically smooth error, SA AMG and CF AMG can each be effective. One benefit of SA AMG is that the method allows for multiple candidate vectors to help define the range of interpolation in order to improve the coarse-grid correction. However, the computational cost of iterating tends to increase substantially when additional candidate vectors are included. Since each candidate vector occupies a column in P for each aggregate (see (3)), additional candidate vectors quickly increase the number of DOFs and nonzeros in coarse-level operators.

As an example, consider a two-level multigrid method for a 2D Laplacian discretized with linear quadrilateral finite elements over a 50×50 uniform grid. In this case, $A_0 \in \mathbb{R}^{2500\times 2500}$, with 21904 nonzero elements. Using the symmetric strength matrix—i.e., $S_{ij} = 1$ if $A_{ij}/\sqrt{A_{ii}A_{jj}} > 0.25$ (see [48])—along with standard, greedy aggregation [48] yields 289 aggregates. This results in $P_0 \in \mathbb{R}^{2500\times 298}$ in the case of a single candidate vector, and a coarse-grid operator $A_1 \in \mathbb{R}^{289\times 289}$ with 2401 nonzeros. Using two candidate vectors, $P \in \mathbb{R}^{2500\times 578}$, and the coarse-grid operator, $A_1 \in \mathbb{R}^{578\times 578}$, has 9604 nonzeros. In this two-grid example, there is approximately a 30% increase in the total number of matrix nonzeros in the hierarchy, which will correspond to a comparable increase in the cost of each iteration.[3] A key feature of the RN AMG method introduced in section 3 is that the growth in complexity is mitigated when incorporating multiple candidate vectors in the range of P.

Additionally, SA AMG provides a simple way to improve interpolation operators

[3] See [30] for an example where a scalar diffusion-like problem requires multiple candidate vectors.

through the interpolation smoothing process. While classical SA AMG uses only one weighted-Jacobi iteration to improve P, multiple iterations as well as smoothing with a *filtered* operator to further improve P have been used [20, 33]. Nevertheless, because each traditional smoothing iteration expands the sparsity pattern of P, this process is limited, a problem which is overcome in RN AMG through a priori fixed sparsity patterns of transfer operators.

2.4. Computational cost. The computational kernel in the multigrid setup and solve phases is a sparse matrix-vector product (SpMV). Thus, a representative measure of the cost of an AMG solver is the number of floating point operations relative to one SpMV with the initial matrix. This measure is referred to as a *work unit* (WU), where one WU is the cost of computing an SpMV on the finest level. Both CF AMG and SA AMG often yield minimal setup costs or setup complexity (SC), but as more features are introduced—e.g., improved SOC methods and energy minimization—the SC may grow. In contrast to the fixed cost of setup, the solve cost depends on the number of iterations or cycles taken, which in turn depends on the stopping residual tolerance. Consequently, the *cycle* complexity (CC), denoted χ_{CC}, is defined as the number of WUs required for each multigrid cycle and is used to measure the solve cost. A similar measure is the *operator* complexity (OC), denoted χ_{OC}, which models the cost of a multigrid hierarchy as the ratio of the total number of nonzeros on all levels to the number of nonzeros on the finest level:

$$\chi_{\text{OC}} = \sum_{\ell} \frac{|A_\ell|}{|A_0|}, \tag{5}$$

where $|C|$ denotes the number of nonzeros in some sparse matrix C. Note that this is equivalent to the total cost of performing one SpMV on each level of the hierarchy. Using this, the CC is often considered to scale with the OC. For example, in the case of a V(2,2) cycle, $\chi_{\text{CC}} \approx 4\chi_{\text{OC}}$. However, a more detailed model for CC includes the residual computation and coarse-grid correction steps. While it is not typical to account for these parts of the solve phase, they often contribute significantly to the CC, especially for the richer interpolation sparsity patterns examined later. To this end, the CC for a V(ν_{pre}, ν_{post}) cycle is defined here as

$$\chi_{\text{CC}} = \sum_{\ell} \frac{(\nu_{\text{pre}} + \nu_{\text{post}} + 1)|A_\ell| + |P_\ell| + |R_\ell|}{|A_0|}, \tag{6}$$

which reflects pre- and postrelaxation, a residual calculation, and one interpolation and restriction per level (see solve phase discussion in section 2).

Detailed estimates of the complexity measures are often neglected in numerical results. One contribution of this work is that precise estimates of the SC, OC, and CC are provided for the numerical results presented in section 5. Coupled with the convergence factor, this information is used to assess the effectiveness of the solver. The SC estimates have been used to expose the expensive parts of the algorithm and motivated the complexity reduction techniques introduced in section 5.1.

3. Root-node method. The general algorithm for constructing an RN AMG hierarchy with $L+1$ levels, using energy-minimizing interpolation smoothing, is given in Algorithm 3 below. The following subsections detail each algorithmic step, comparing and contrasting with the SA AMG and CF AMG.

Algorithm 3: `root_node_setup`

Input: A_0: fine-grid operator
B_0: fine-grid candidate vectors for A
$\hat{B}_0$: fine-grid candidate vectors for A^T (if $A \neq A^T$)
d: interpolation sparsity pattern width
`vector`: flag indicating vector-based problem
`max_size`: threshold for max size of coarsest problem
`prefilter`: prefiltering of interpolation sparsity pattern
`postfilter`: postfiltering of interpolation sparsity pattern

Output: $A_1, \ldots, A_L$,
$P_0, \ldots, P_{L-1}$,
$R_0, \ldots, R_{L-1}$

$\ell = 0$
while `size`$(A_\ell) >$ `max_size`
  $S_\ell =$ `strength`(A_ℓ) {strength-of-connection of matrix}
  if `vector`
    $S_\ell =$ `amalgamate`(S_ℓ) {amalgamate from degree-of-freedom to nodal}
  C_ℓ, roots = `aggregate`(S_ℓ) {construct aggregates and root-nodes}
  $N_\ell \leftarrow S_\ell^d C_\ell$ {form interpolation sparsity pattern}
  if `prefilter`
    $N_\ell =$ `filter`(N_ℓ) {eliminate small entries}
  if `vector`
    N_ℓ, roots = `unamalgamate`$(N_\ell$, roots) {from nodal to degree-of-freedom}
  $N_\ell =$ `root_node_pattern`(N_ℓ) {convert to root-node pattern}
  $B_\ell =$ `smooth`(A_ℓ, B_ℓ) {improve candidates with relaxation}
  $T_\ell, B_{\ell+1} =$ `inject`$(C_\ell, N_\ell, B_\ell$, roots) {form tentative interpolation and $B_{\ell+1}$}
  $P_\ell =$ `improve`$(A_\ell, T_\ell, B_\ell, B_{\ell+1})$ {create smooth P_ℓ with $P_\ell B_{\ell+1} = B_\ell$}
  if `postfilter`
    $P_\ell =$ `filter`(P_ℓ) {eliminate small entries}
    $P_\ell =$ `enforce`$(P_\ell, B_\ell, B_{\ell+1})$ {enforce mode constraint with (12)}
    $P_\ell =$ `improve`$(A_\ell, P_\ell, B_\ell, B_{\ell+1})$ {re-smooth P with one iteration}
  if `symmetric`(A_ℓ)
    $R_\ell = P_\ell^T$
  else
    $\hat{B}_\ell =$ `smooth`$(A_\ell^T, \hat{B}_\ell)$ {improve candidates for the nonsymmetric case}
    $\hat{T}_\ell, \hat{B}_{\ell+1} =$ `inject`$(C_\ell, N_\ell, \hat{B}_\ell$, roots) {form tentative restriction and $\hat{B}_{\ell+1}$}
    $R_\ell^T =$ `improve`$(A_\ell^T, \hat{T}_\ell, \hat{B}_\ell, \hat{B}_{\ell+1})$ {create smooth R_ℓ with $R_\ell^T \hat{B}_{\ell+1} = \hat{B}_\ell$}
    if `postfilter`
      $R_\ell^T =$ `filter`(R_ℓ^T) {eliminate small entries}
      $R_\ell^T =$ `enforce`$(R_\ell^T, \hat{B}_\ell, \hat{B}_{\ell+1})$ {enforce mode constraint with (12)}
      $R_\ell^T =$ `improve`$(A_\ell^T, R_\ell^T, \hat{B}_\ell, \hat{B}_{\ell+1})$ {re-smooth R with one iteration}
  $A_{\ell+1} = R_\ell A_\ell P_\ell$ {form coarse-grid}
  $\ell = \ell + 1$

3.1. Candidate vectors B. As with SA AMG, a priori knowledge of the algebraically smooth error is assumed as input in the form of a set of candidate vectors B. These vectors are critical for ensuring accurate interpolation of important algebraically smooth modes. In the case of A being symmetric, one set of candidate vectors, B, is sufficient. In the case of nonsymmetric problems, a restriction operator R is formed independently (whereas $R = P^T$ in the case of symmetry) through the *left* candidate vectors $\hat{B}$, which target smooth error in A^T. Generally if candidate vectors are not known or provided, a constant vector is used, as the constant is geometrically smooth and a good choice for many problems. Finally, to ensure smooth (including at the boundaries) candidate vectors, a small number of relaxation sweeps are applied to $A_\ell B_\ell = \mathbf{0}$ (line 13). This improves the algebraic smoothness of B_ℓ, especially near boundaries of a domain. Even for textbook examples such as a Laplacian, the standard candidate $B_\ell = \mathbf{1}$ can yield a poor approximation to algebraically smooth error near Dirichlet boundaries.

3.2. Strength matrix S. The first level-specific step is the construction of an $n_\ell \times n_\ell$ SOC matrix, S_ℓ (line 3), which indicates strong connections between DOFs in the problem. This matrix is used for the aggregation of DOFs, and for the construction of a sparsity pattern, $\mathcal{N}$, for P. This work considers the classical strength measure [42], symmetric strength measure [48], and so-called evolution strength measure [39], although other measures have also been proposed [10, 12]. The classical and symmetric strength measures essentially look at the magnitude of an off-diagonal entry when determining whether it represents a strong connection. In contrast, the evolution measure computes strength around a DOF i by locally evolving a unit vector centered at i with a few sweeps of weighted-Jacobi relaxation. This creates a locally smooth vector which is then postprocessed to determine which matrix entries in row i are strong. For instance, for anisotropic diffusion, the directions in which the unit vector diffuses most quickly are selected as strong connections.

After finishing, each method produces a matrix S_ℓ in which individual entries represent the SOC in the graph of A_ℓ. One modification to S_ℓ that is used here is to normalize each row so that

$$\max_{j \neq i} (S_\ell)_{ij} = (S_\ell)_{ii} = 1. \tag{7}$$

That is, all elements are nonnegative, and for each row in S_ℓ the diagonal and the largest off-diagonal, i.e., the strongest connection, both equal 1. This scaling and the nonbinary nature of S_ℓ are important when computing the sparsity pattern in line 7.

For vector-based problems, A_ℓ has a block structure of block-size $m \times m$, and for common cases such as elasticity each block corresponds to the DOFs associated with different variables but defined at the same spatial node. It is typical to group each block into a single so-called *supernode* [48], followed by aggregation only at the supernode level. This requires condensing the SOC with an `amalgamate` step in line 5, which reduces S_ℓ to an $n_\ell/m \times n_\ell/m$ matrix. The amalgamated entry is equal to the maximal entry of its associated $m \times m$ block in S_ℓ. This allows for aggregation based on supernodes.

3.3. Aggregation. The next operation is forming an aggregation, $\mathcal{A}_\ell$, as in SA AMG (see Algorithm 1), and an associated list of root-nodes (line 6). The classical greedy aggregation algorithm [48] is used, wherein an unaggregated vertex in S_ℓ is selected (as the root-node) and all neighboring vertices with strong edge weights are collected to form an aggregate. An aggregation pattern matrix C_ℓ is then defined as

a partition of unity such that $C_\ell(i,j) = 1$ if node i belongs to aggregate j, and zero otherwise. This pattern will be used in forming the sparsity pattern for P_ℓ. Other aggregation routines such as pairwise aggregation [22, 36] have been considered but have not demonstrated improvements in RN AMG performance, and the choice of root-node is less clear than in a greedy aggregation routine.

Example aggregates for standard isotropic diffusion problems are given in Figure 3. For the systems case, m DOFs at each supernode implies that each root-node also contains m DOFs.

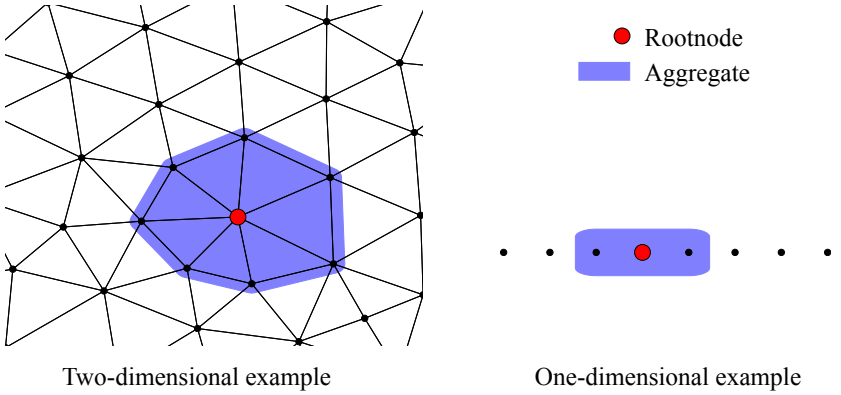


FIG. 3. *Sample aggregates on the finest level for standard isotropic diffusion in one and two dimensions.*

3.4. Arbitrary sparsity pattern $\mathcal{N}$. Aggregation gives an interpolation structure, with each root-node corresponding to one block column of T and P as in (3), but not a sparsity pattern for P. Thus, in lines 7–12, the sparsity pattern $\mathcal{N}_\ell$ for interpolation is built, where $(i,j) \notin \mathcal{N}_\ell \Leftrightarrow N_\ell(i,j) = 0$. Nonzero elements are based on growing the aggregation pattern matrix, C_ℓ, based on the strength matrix, S_ℓ, through multiplication $N_\ell = S_\ell^d C_\ell$ for some *degree* or *distance* d. Using d applications of S_ℓ extends the interpolation stencil for a given F-point to a distance of d in the strength matrix. Large values of d allow the sparsity pattern to grow in the direction of strong connections, allowing for long-distance interpolation. This approach differs from that of [41, 45] in that a normalized SOC matrix (7) is used with C_ℓ, and as a result, N_ℓ can be *filtered* (line 9) by examining the magnitude of the entries: larger entries indicate a stronger path from the root-node i. As a result, the normalization of S_ℓ imbues a relative size across columns, so that the product with the binary aggregation matrix C_ℓ yields individual entries related to the SOC.

In the case of vector problems, the `unamalgamate` step in line 11 reverses amalgamation and converts the list of root-nodes into a list of DOFs. If the amalgamated list of roots is of length $n_{\ell+1}/m$, then the unamalgamated list is of length n_ℓ. Similarly, the unamalgamated entry $N_\ell(i,j)$ is equal to the amalgamated entry $N_\ell(\lfloor i/m \rfloor, \lfloor j/m \rfloor)$.

The last step in computing the sparsity is `root_node_pattern` in line 12. This function enforces the root-node pattern of interpolation in (4) by traversing N_ℓ to change each root-node row to be the corresponding row of the identity—i.e., the restriction to each root-node for the coarse-grid does not involve other root-nodes, and each root-node is interpolated by value back to the fine-grid. Other than this root-node requirement, RN AMG allows for arbitrary sparsity patterns and enables selective control of the number of interpolation points. This flexibility is used in the

next section, where the sparsity pattern is filtered (dropping entries) in a way that targets only *strong* long-distance connections in P_ℓ.

It should be noted that the sparsity pattern for R^T is built in the same manner as P: by expanding a tentative operator with an SOC matrix based on A. Another option is to construct the sparsity pattern for R based on an SOC from A^T; however, numerical results consistently indicate a degradation in convergence factors when doing the latter. Choosing the *optimal* sparsity pattern for both P and for R remains an important and open research question.

3.5. Filtering sparsity pattern. A filtering step, which removes nonzeros in N_ℓ, is used after the construction of the sparsity pattern for P. A large degree d (see line 7) is often needed [30, 45] to construct effective interpolation operators; filtering can limit the additional cost due to the growth in the sparsity pattern. In particular, filtering allows for long-distance interpolation in the direction of strong connections, while limiting complexity.

Prefiltering is used to filter the sparsity pattern matrix N_ℓ before the energy-minimization. Here, entries are eliminated in N_ℓ prior to constructing P_ℓ based on the size of entries in $N_\ell = S_\ell^d C_\ell$, which indicate the SOC between two DOFs. Because prefiltering is only based on SOC and not the fully formed interpolation operator, it is possible that influential entries are inadvertently removed, thus degrading convergence. However, in practice, trimming the sparsity pattern of P_ℓ before initiating the construction significantly lowers the SC in many cases, with minimal impact on AMG convergence.

Given an initial sparsity pattern, entries are filtered as in [23], by either retaining the k largest values in a row or by applying a drop tolerance θ. Algorithm 4 describes this process in detail, where $\max(G, i, k)$ is the kth largest off-diagonal entry in row i. The idea of prefiltering has shown to be effective for model problems using a polynomial approximation to A_{ff}^{-1} in [14]. In contrast, the prefiltering used here is less expensive and relies on values already computed by the root-node algorithm.

Algorithm 4: `filter`(G)

Input: G: matrix to be filtered
θ: filtering drop-tolerance
k: filtering threshold

Output: G

```
1 if k
2 |  for |G(i,j)| < max(G,i,k) do
3 |  |  G(i,j) ← 0
4 if θ
5 |  for |G(i,j)| < θ max(G,i,1) do
6 |  |  G(i,j) ← 0
```

3.6. Interpolation construction. A *tentative* interpolation operator, T, is constructed from the sparsity pattern. The full interpolation is then formed based on a constrained energy-minimization with the following principles:

i. T and P satisfy interpolation constraints of the provided algebraically smooth candidates. That is,

$$TB_c = B \quad \text{and} \quad PB_c = B \tag{8}$$

for candidates B and coarse level candidates B_c.

ii. The `improve` procedure reduces the *energy* of each jth column of P:

$$\|P_{(j)}\| \leq \|T_{(j)}\| \tag{9}$$

for some A-induced norm $\|\cdot\|$. Interpolation smoothing in SA AMG is one example.

iii. Given sparsity pattern $\mathcal{N}$,

$$T_{ij} = 0 \quad \text{if } (i,j) \notin \mathcal{N} \quad \text{and} \quad P_{ij} = 0 \quad \text{if } (i,j) \notin \mathcal{N}. \tag{10}$$

The energy-minimization approach [41] is used here, which satisfies these principles. Energy-minimization is an iterative smoothing process that improves P through several passes. As a result, growth in the sparsity pattern of P necessitates a constraint on the sparsity pattern constraint, $\mathcal{N}$. However, in enforcing $\mathcal{N}$ by dropping entries in P, the constraints are no longer satisfied. In response, the constraints are enforced as an additional step.

RN AMG proceeds by taking the aggregation and list of root-nodes to construct the coarse-grid candidates by injection, $B_{\ell+1}(i,j) = B_\ell(k,j)$, where k is the ith root-node. If m is the block-size of the original matrix ($m = 1$ for a scalar problem), then *only* the first m candidates are injected over each aggregate to form an initial T_ℓ (line 14). As a result, each root-node represents m DOFs on the coarse-grid. An additional step is performed on T_ℓ, normalizing each column so that the coarse-grid variables are interpolated by value to the fine-grid root-nodes. This process yields the following form:

$$T_\ell = \begin{bmatrix} W_\ell \\ I \end{bmatrix} \begin{matrix} \} \text{ non–root-nodes,} \\ \} \text{ root-nodes.} \end{matrix} \tag{11}$$

For $m = 1$, T_ℓ has nonoverlapping columns; for $m > 1$, W_ℓ is block-diagonal, as T in (3). With the identity over C-points in T_ℓ, interpolation in RN AMG resembles that of CF AMG (cf. (4)).

If there are more than m candidates, the remaining candidates are projected into $\text{ran}(T_\ell)$ in the Euclidean inner-product. In RN AMG, it is assumed that the sparsity pattern has sufficient DOFs that this is an underdetermined problem. This results in the minimal norm update to each row of T_ℓ such that $T_\ell B_{\ell+1} = B_\ell$ and T_ℓ obeys the sparsity pattern $\mathcal{N}_\ell$. More specifically, an update u to only the allowed nonzero portion of row i of T_ℓ (called t) is computed by solving

$$(t+u)B_{\ell+1} = B_\ell \quad \Leftrightarrow \quad B_{\ell+1}^T u^T = B_\ell^T - B_{\ell+1}^T t^T \tag{12}$$

for u using least-squares. Note that only W_ℓ is modified in (12); that is, injection over C-points is maintained, and W_ℓ is expanded to interpolate candidate vectors.

Next, interpolation P_ℓ is formed using `improve` in line 15. Energy-minimization forms P_ℓ as a succession of energy-minimization updates to T_ℓ. Each update U is computed to reduce the energy of each column via a Krylov process and also to satisfy $UB_{\ell+1} = 0$. As a result, (i) $(T_\ell + U)B_{\ell+1} = T_\ell B_{\ell+1}$ (constraints are satisfied exactly), and (ii) $\|T_\ell + U\| \leq \|T - \ell\|$ (energy is reduced). The projection operator enforcing the constraints is analogous to (12) (see [41]).

3.7. Filtering interpolation. After P_ℓ is formed, a postfiltering process (similar to the prefiltering in section 3.5) is applied in line 17 to reduce complexity. Postfiltering removes elements directly from P_ℓ after smoothing, but this leads to a P_ℓ that

violates the mode interpolation constraints (8). Thus, the function `enforce` is used to reapply these constraints via (12). Finally, an additional iteration of `improve` is used to account for large increases in energy caused by removing entries. One advantage of postfiltering is that element removal is based on the actual smoothed interpolation stencil entries, and is thus less likely to inadvertently degrade convergence compared with prefiltering—i.e., by removing important entries from P. Postfiltering generally results in a lower complexity in the Galerkin coarse-grid operator (line 30) and all subsequent coarser grid operations. A similar filtering approach is also effective for CF AMG methods [23]. However, postfiltering does not reduce the OC and SC as effectively as prefiltering, in particular because energy-minimization (one of the dominant costs in SC) is applied to a larger sparsity pattern, which is then trimmed afterwards.

3.8. Coarse-grid construction. The final step in Algorithm 3 is the construction of the coarse-grid operator through a (Petrov) Galerkin triple-matrix product $A_{\ell+1} = R_\ell A_\ell P_\ell$. For symmetric A_ℓ, restriction is the transpose of interpolation. For nonsymmetric A_ℓ, the interpolation construction process is duplicated for R_ℓ^T using A_ℓ^T (the $\hat{\cdot}$ notation is used to denote the quantities used to compute R_ℓ) [18].

3.9. Discussion: Compare and contrast RN AMG with SA AMG. Similarly to SA AMG, RN AMG facilitates the use of multiple arbitrary candidate vectors, but it handles the complexity challenges by not adding columns to P for each additional candidate. The dimensions of T and P are fixed, with the number of columns in T and P being equal to the number of aggregates. The candidates B are projected exactly into ran(P) and ran(T), assuming a sparsity pattern with enough entries. With SA AMG, for each candidate vector added to B, a new column is added to T and P for every aggregate (see (3)). This difference allows RN AMG to exhibit significantly lower complexity than SA AMG in some instances.

RN AMG also uses interpolation smoothing, like SA AMG, so that P is iteratively improved. However, standard interpolation smoothing does *not* satisfy the exact candidate interpolation constraints in general, although it does attempt to target the same three principles outlined in section 3.6 (including accurate but not exact candidate vector interpolation). A consequence of satisfying the constraints is that the candidate vectors are exactly represented in the range of interpolation. To measure this, consider the error in interpolating B found through an orthogonal projection of B into the range of interpolation:

$$e_B = (I - PP^\dagger)B, \tag{13}$$

where $P^\dagger$ is the pseudoinverse of P. The pointwise values of the error e_B are shown in Figure 4, where RN AMG achieves much lower error, directly satisfying the constraints (globally).

Additionally, standard interpolation smoothing does not allow for multiple smoothing passes (and hence longer-distance interpolation) without suffering from complexity issues and fill-in in P. The arbitrary sparsity pattern allows RN AMG to have longer-distance interpolation and multiple smoothing iterations, while also effectively managing complexity.

An SA AMG-type method supporting vector problems can be derived from Algorithm 3 by removing the root-node specific lines and by using the `inject` and `smooth` functions from Algorithm 1 to form T_ℓ and P_ℓ. This form of SA AMG also supports nonsymmetric problems [43].

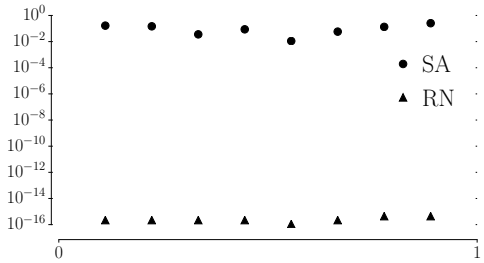
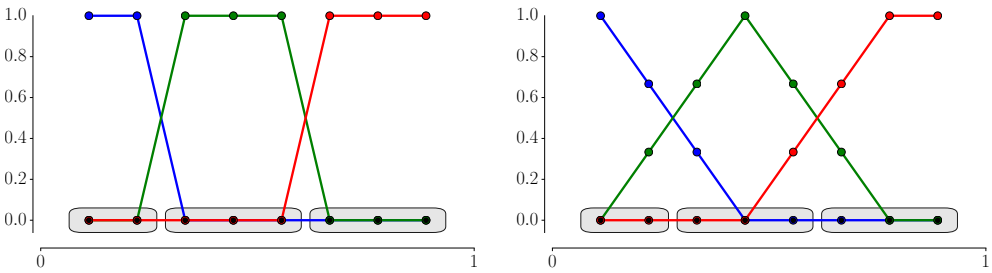


FIG. 4. *Error e_B in satisfying the constraints $B \equiv \mathbf{1} \in \text{ran}(P)$ for a 1D Poisson problem with eight DOFs using central finite differences.*

3.10. Discussion: Compare and contrast RN AMG with CF AMG. Compared with CF AMG, RN AMG adopts the structure of P from (4), which aids the use of pre- and postfiltering since the columns of interpolation are normalized (with the identity). (In section 4, additional theoretical motivation for using this form is investigated.) To understand this, consider how the CF AMG structure (11) scales interpolation around each root-node for a simple 1D Laplace example on eight nodes with finite differencing and $B = \mathbf{1}$. In this case, three aggregates are formed, as in Figure 2(a). Each column of T is shown in Figure 5(a), where a relative balance in weighting across aggregates is observed, in contrast to SA AMG, which is shown in Figure 2(b). This balance in weighting is evident again in the final P in Figure 5(b), again because the identity form is preserved according to (11).



(a) Columns of tentative interpolation T. (b) Columns of interpolation P.

FIG. 5. *Interpolation using a single candidate $B = \mathbf{1}$ in RN AMG.*

This scaling aids the filtering strategy, which eliminates relatively small interpolation weights in the rows of P, because it provides the rationale for comparing the magnitude of entries across columns. In other words, RN AMG stipulates that a large interpolation weight is one, representing injection from the coarse- to fine-grid, visible as the peaks in Figure 5(b).

In contrast to CF AMG, there are the many similarities that RN AMG shares with SA AMG, with key differences being the ability to iteratively improve P and guarantee accurate interpolation of the user-defined candidate vectors B. Regarding similarities to more recent adaptive CF-style AMG methods, bootstrap AMG (BAMG) [8] also fits multiple candidate vectors into the range of interpolation but does so by generating a set of many candidates and overdetermining each row of P. In this respect, RN AMG is closer to SA AMG, as each row of P is underdetermined and fits only a small

number of candidate vectors into the range of P, using interpolation smoothing to fully determine each interpolation entry.

In contrast to both CF AMG and SA AMG, RN AMG allows for automatically expanded sparsity patterns $\mathcal{N}_\ell$, which can be filtered to facilitate long-distance interpolation. This is critical for robustness and performance, as shown in the examples in section 5, and is a unique feature of RN AMG.

4. Theoretical motivation for root-node. In this section, theoretical motivation for RN AMG is introduced. Initially, it is assumed that A is SPD, in order to connect with classical AMG theory. Based on relations established in the symmetric setting, some results are extended to nonsymmetric systems in section 4.2.

4.1. The symmetric case. Let $A \in \mathbb{R}^{n\times n}$ be SPD; $\|\cdot\|$ and $\|\cdot\|_A$ represent the l^2- and A-norms, respectively; and $P : \mathbb{R}^{n_c} \to \mathbb{R}^n$ be an interpolation operator defining a coarse space of size n_c. The error propagation operator for a two-grid method is given by

$$E_{\mathrm{TG}} = I - P(P^TAP)^{-1}P^TA, \tag{14}$$

and a multilevel version E_{MG} is similarly defined (see [32]). Here, bounds on the A-norm of E_{TG} and E_{MG} are constructed. The *weak approximation property* (WAP) gives necessary and sufficient conditions for two-grid convergence as follows: there exists $K \in \mathbb{R}$ such that for any vector $\mathbf{u} \in \mathbb{R}^n$,

$$\min_{\mathbf{w}_c\in\mathbb{R}^{n_c}} \|\mathbf{u} - P\mathbf{w}_c\|^2 \le \frac{K}{\|A\|}\|\mathbf{u}\|_A^2, \tag{15}$$

wherein $\|E_{TG}\|_A = 1 - \frac{1}{K_{TG}}$ and $1 \le K_{TG} \le K$ [49]. For simplicity, (15) is based on Richardson relaxation. A WAP with respect to a general relaxation scheme along with a tight bound on K_{TG} can be found in [25, 49]. The *strong approximation property* (SAP) establishes multilevel convergence with a stronger condition: there exists $K \in \mathbb{R}$ such that for any vector $\mathbf{u} \in \mathbb{R}^n$,

$$\min_{\mathbf{w}_c\in\mathbb{R}^{n_c}} \|\mathbf{u} - P\mathbf{w}_c\|_A^2 \le \frac{K}{\|A\|}\|A\mathbf{u}\|^2. \tag{16}$$

If (16) holds on each level of the hierarchy, then $\|E_{MG}\|_A = 1 - \frac{1}{K_{MG}}$ and $1 \le K_{MG} \le 1 + K\frac{\|M\|}{\|A\|}$, where M is the chosen relaxation scheme of the form $\mathbf{x}_{k+1} = \mathbf{x}_k + M^{-1}\mathbf{r}_k$ for residual $\mathbf{r}$ [49].

Since A is assumed to be SPD, its eigenvectors form an l^2- and A-orthonormal basis for the space $\mathbf{R}^n$. Thus, if the WAP and SAP hold for all eigenvectors, they hold for all vectors, and it follows that the WAP requires that eigenvectors be interpolated with accuracy on the order of the corresponding eigenvalue, and the SAP requires that interpolation accuracy on the order of the eigenvalue squared. This leads to an equivalence of satisfying the WAP based on A^2 and the SAP for A as follows.

LEMMA 4.1 (Lemma 5.20 of [49]). *Let $A \in \mathbb{R}^{n\times n}$ be SPD and $P \in \mathbb{R}^{n\times n_c}$. Then*

$$\min_{\mathbf{w}_c\in\mathbb{R}^{n_c}} \|\mathbf{u} - P\mathbf{w}_c\|^2 \le \frac{K^2}{\|A^2\|}\|\mathbf{u}\|_{A^2}^2 \quad \textit{for all } \mathbf{u} \tag{17}$$

if and only if

$$\min_{w_c} \|\mathbf{u} - P\mathbf{w}_c\|_A^2 \le \frac{K}{\|A\|}\|A\mathbf{u}\|^2 \quad \textit{for all } \mathbf{u}. \tag{18}$$

The accuracy demands of the WAP and SAP with respect to eigenvalues indicates that the range of P should contain eigenvectors of A associated with small eigenvalues (or so-called *algebraically smooth* modes). In building AMG hierarchies, this generally takes one of two forms, (i) ensuring that known low-energy modes are exactly represented in the range of P, and (ii) minimizing columns of P in the A-norm so that the range of P corresponds to algebraically smooth vectors. Recall that RN AMG combines each of these approaches in a constrained energy-minimization [14, 41, 45, 40].

Given a CF-splitting of the current grid, consider a matrix ordering of the form $A = \left[\begin{smallmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{smallmatrix}\right]$, where A_{ff} corresponds to F-point-to-F-point connections, A_{fc} to F-point-to-C-point connections, and so on. In a CF AMG context, interpolation is then assumed to have the form $P = \left(\begin{smallmatrix} W \\ I \end{smallmatrix}\right)$, where the lower identity block interpolates and restricts C-points by injection and W interpolates and restricts F-points based on linear combinations. The minimizing coarse-grid vector, $\mathbf{w}_c$, in the WAP and the SAP is given by l^2-orthogonal and A-orthogonal projections, respectively, of the vector $\mathbf{u}$ onto the range of P. In practice, such projections are generally too expensive to form explicitly; thus, computable measures are also of interest. One option consistent with CF AMG is to let $\mathbf{w}_c = \mathbf{u}_c$, that is, define $\mathbf{w}_c$ as the restriction of $\mathbf{u}$ to C-points. This provides a bound on the WAP, as $\min_{\mathbf{w}_c \in \mathbb{R}^{n_c}} \|\mathbf{u} - P\mathbf{w}_c\|^2 \leq \|\mathbf{u} - P\mathbf{u}_c\|^2$ for all $\mathbf{u}$, and thus

$$\mu(P) := \max_{\mathbf{u} \neq 0} \frac{\|\mathbf{u} - P\mathbf{u}_c\|^2}{\|\mathbf{u}\|_A^2} \geq K_{TG}. \tag{19}$$

Assuming $P = \left(\begin{smallmatrix} W \\ I \end{smallmatrix}\right)$, the optimal interpolation operator under $\mu(P)$ is given by

$$\begin{aligned} P_{\text{ideal}} &= \operatorname*{argmin}_P \max_{\mathbf{u}} \frac{\|\mathbf{u} - P\mathbf{u}_c\|^2}{\|\mathbf{u}\|_A^2} \\ &= \begin{bmatrix} -A_{ff}^{-1} A_{fc} \\ I \end{bmatrix}, \end{aligned} \tag{20}$$

where P_{ideal} is referred to as "ideal interpolation."

In addition to P_{ideal} being optimal with respect to the measure $\mu(P)$ (and thus satisfying the WAP for some $K \geq K_{TG}$), if A_{ff} is well-conditioned, then $\|A_{ff}^{-1}\|$ is bounded by a small constant, in which case P_{ideal} also satisfies the following SAP:

$$\min_{\mathbf{w}_c \in \mathcal{V}_c} \|\mathbf{u} - P_{\text{ideal}}\mathbf{w}_c\|_A^2 \leq \|\mathbf{u} - P_{\text{ideal}}\mathbf{u}_c\|_A^2 \leq \|A_{ff}^{-1}\| \|A\mathbf{u}\|^2 \leq \frac{K}{\|A\|} \|A\mathbf{u}\|^2$$

for some K. While P_{ideal} indicates an effective interpolation scheme, $A_{ff}^{-1} A_{fc}$ is often a dense matrix and difficult to compute. However, if A_{ff} is well-conditioned, its entries decay exponentially fast away from the diagonal [14], suggesting that a sparse approximation can be formed. In addition to directly satisfying constraint vectors, the energy-minimization piece of root-node interpolation constructs a sparse approximation to P_{ideal}, with the goal of retaining the convergence properties of ideal interpolation while limiting coarse-grid complexity. In general, aggregation-based AMG is motivated through energy-minimization principles over the columns of the interpolation operator [19, 21, 29, 35, 46, 47]. The root-node approach, however, is supported by a more complete theoretical motivation, as well as practical benefits including the use of aggregation or CF-splittings for coarsening, and better conditioning of the coarse-grid operator in the nonsymmetric setting (section 4.2).

Observe that $AP_{\text{ideal}} = \left(\begin{smallmatrix} 0 \\ S \end{smallmatrix}\right)$, where $S = A_{cc} - A_{cf}A_{ff}^{-1}A_{fc}$ is the Schur complement of A. Given that $AP_{\text{ideal}} = 0$ over F-points, this motivates minimizing columns of $P = \left(\begin{smallmatrix} W \\ I \end{smallmatrix}\right)$ in the A-norm to approximate the action of P_{ideal}. The identity block over C-points along with any constraints enforced ensure that columns of W are nonzero. (The solution to minimizing a general P in the A-norm without constraints is $P = \mathbf{0}$.) Coupled with a predetermined sparsity pattern and constraints $PB_c = B$, where B is a set of columnwise constraint vectors to be in the range of P and B_c B restricted to C-points, energy minimization in the A-norm is exactly the conjugate gradient (CG) variant of energy-minimization proposed in [41]. Lemma 4.2 shows the relationship between P_{ideal} and energy-minimization. That is, the CG variant of energy-minimization used to form P over a given sparsity pattern is equivalent to minimizing the difference between columns of P and P_{ideal} in the A-norm over a given sparsity pattern.

LEMMA 4.2. *Let $A \in \mathbb{R}^{n\times n}$ be SPD, $P_{\text{ideal}} \in \mathbb{R}^{n\times n_c}$ be given by (20), and $\mathbf{e}_\ell$ be the ℓth canonical basis vector, where $\mathbf{p}_\ell = P\mathbf{e}_\ell$ is the ℓth column of P. Denote by $\mathcal{N}^F$ a sparsity pattern for any matrix $W \in \mathbb{R}^{n_f\times n_c}$, where $W_{ij} \neq 0$ if $(i,j) \in \mathcal{N}^F$. Then define the set $\mathcal{P}_\ell$ as the ℓth column of any matrix with the structure of P_{ideal}, restricted to sparsity pattern $\mathcal{N}^F$; that is,*

$$\mathcal{P}_\ell = \left\{ P\mathbf{e}_l \,:\, P = \begin{bmatrix} W \\ I \end{bmatrix}, \text{where } W \in \mathbb{R}^{n_f\times n_c} \text{ and } W_{ij} = 0 \text{ if } (i,j) \notin \mathcal{N}_{ij}^F \right\}. \tag{21}$$

Then for $l = 1, \ldots, n_c$,

$$\underset{\mathbf{p}_\ell\in\mathcal{P}_\ell}{\operatorname{argmin}} \|\mathbf{p}_\ell\|_A = \underset{\mathbf{p}_\ell\in\mathcal{P}_\ell}{\operatorname{argmin}} \|\mathbf{p}_\ell - P_{\text{ideal}}\mathbf{e}_\ell\|_A \,. \tag{22}$$

Equivalently, minimizing the columns of P in the A-norm is equivalent to minimizing the difference between columns of P and P_{ideal} in the A-norm.

Proof. Equivalence is established by demonstrating identical weak forms for the two minimization problems in (22). Consider $\mathbf{p}_\ell \in \mathcal{P}_\ell$, and define N_ℓ to be the diagonal matrix that enforces sparsity pattern $\mathcal{N}^F$ on the ℓth column of W, $\mathbf{w}_\ell$. That is, $\mathbf{w}_\ell = W\mathbf{e}_\ell = N_\ell W\mathbf{e}_\ell$, where the kth entry of $\mathbf{w}_\ell$ equals zero if $(k,\ell) \notin \mathcal{N}^F$.

1. Consider minimizing the lth column of P, given by $\mathbf{p}_\ell = \left[\begin{smallmatrix} \mathbf{w}_\ell \\ \mathbf{e}_\ell \end{smallmatrix}\right]$, in the A-norm. To this end, define the functional $G(\mathbf{w}_\ell) = \left\langle A\mathbf{p}_\ell, \mathbf{p}_\ell \right\rangle = \left\langle A\left[\begin{smallmatrix} \mathbf{w}_\ell \\ \mathbf{e}_\ell \end{smallmatrix}\right], \left[\begin{smallmatrix} \mathbf{w}_\ell \\ \mathbf{e}_\ell \end{smallmatrix}\right] \right\rangle$, with first variation
$$G'(\mathbf{w}_\ell; \mathbf{v}) = 2\Big\langle N_\ell A_{ff} N_\ell \mathbf{w}_\ell - N_\ell A_{fc}\mathbf{e}_\ell, \mathbf{v} \Big\rangle$$
for $\mathbf{v} = N_\ell \mathbf{v}$. The weak form for minimizing G is then given by
$$N_\ell A_{ff} N_\ell \mathbf{w}_\ell = N_\ell A_{fc}\mathbf{e}_\ell.$$
2. Consider minimizing the difference between the lth column of P and P_{ideal} in the A-norm. That is, define the functional
$$\begin{aligned} H(\mathbf{w}_\ell) &= \Big\langle A(P - P_{\text{ideal}})\mathbf{e}_\ell, (P - P_{\text{ideal}})\mathbf{e}_\ell \Big\rangle \\ &= \Big\langle A_{ff}(N_\ell \mathbf{w}_\ell - A_{ff}^{-1}A_{fc}\mathbf{e}_\ell), N_\ell \mathbf{w}_\ell - A_{ff}^{-1}A_{fc}\mathbf{e}_\ell \Big\rangle. \end{aligned}$$
Taking the first variation yields
$$H'(\mathbf{w}_\ell; \mathbf{v}) = \Big\langle N_\ell A_{ff} N_\ell \mathbf{w}_\ell - N_\ell A_{fc}\mathbf{e}_\ell, \mathbf{v} \Big\rangle,$$

with $\mathbf{v} = N_\ell \mathbf{v}$, resulting in the weak form

$$N_\ell A_{ff} N_\ell \mathbf{w}_\ell = N_\ell A_{fc} \mathbf{e}_\ell. \qquad \square$$

Remark 4.3. A result similar to Lemma 4.2 can be found in [14] in the Frobenius norm.

Although P_{ideal} in (20) is motivated through (and optimal in the sense of (19)) the WAP (15), and with an appropriate CF-splitting satisfies the SAP (16), P_{ideal} is not optimal in any sense with respect to the SAP. However, a similar derivation leads to an equivalent "ideal interpolation" operator with respect to the SAP, as introduced in Lemma 4.4.

LEMMA 4.4. *Let $A \in \mathbb{R}^{n\times n}$ be SPD and $P \in \mathbb{R}^{n\times n_c}$ take the form $P = \begin{pmatrix} W \\ I \end{pmatrix}$; that is, C-points are interpolated by injection. Then, consider the SAP under the assumption that the preimage of any vector $\mathbf{u}$ under P is given by $\mathbf{u}_c$,*

$$\hat{\mu}(P) := \max_{\mathbf{u}\neq 0} \frac{\|\mathbf{u} - P\mathbf{u}_c\|_A^2}{\|A\mathbf{u}\|^2}.$$

Then for any smoothing scheme M, $K_{MG} \leq 1 + \hat{\mu}(P)\frac{\|M\|}{\|A\|}$, and the optimal P with respect to minimizing $\hat{\mu}$ is given by

$$\operatorname*{argmin}_P \hat{\mu}(P) = \begin{bmatrix} (A_{ff}^2 + A_{fc}A_{cf})^{-1}(A_{ff}A_{fc} + A_{fc}A_{cc}) \\ I \end{bmatrix},$$

which is exactly ideal interpolation (20) *for A^2.*

Proof. First note from the SAP that

$$\begin{aligned} K_{MG} &\leq 1 + \left(\max_{\mathbf{u}\neq 0} \min_{\mathbf{w}_c} \frac{\|\mathbf{u} - P\mathbf{w}_c\|_A^2}{\|A\mathbf{u}\|^2} \right) \frac{\|M\|}{\|A\|} \\ &\leq 1 + \left(\max_{\mathbf{u}\neq 0} \frac{\|\mathbf{u} - P\mathbf{u}_c\|_A^2}{\|A\mathbf{u}\|^2} \right) \frac{\|M\|}{\|A\|}. \end{aligned}$$

Based on the proofs of Theorem 3.1 and Corollary 3.2 in [25], it follows that

$$\operatorname*{argmin}_P \max_{\mathbf{u}\neq 0} \frac{\|\mathbf{u} - P\mathbf{u}_c\|_A^2}{\|A\mathbf{u}\|^2} = \operatorname*{argmin}_P \max_{\mathbf{u}\neq 0} \frac{\|\mathbf{u} - P\mathbf{u}_c\|^2}{\|A\mathbf{u}\|^2} = \operatorname*{argmin}_P \max_{\mathbf{u}\neq 0} \frac{\|\mathbf{u} - P\mathbf{u}_c\|^2}{\|\mathbf{u}\|_{A^2}^2}.$$

The final equation is $\mu(P)$ from (19) as applied to A^2, and thus the minimum is attained by ideal interpolation (20) as applied to A^2. $\square$

Remark 4.5. An equivalent result holds for A^*A and $\sqrt{A^*A}$ (as opposed to A^2 and A), which is used in the following section on nonsymmetric operators.

4.2. The nonsymmetric case. The difficulties of convergence theory for nonsymmetric A lie in the fact that A no longer defines a norm, raising the question of an appropriate norm for measuring convergence. The spectral radius bounds asymptotic convergence factors [37, 50]; however, consider the following error propagation matrix as an example:

$$E = \begin{pmatrix} \epsilon & k \\ 0 & \epsilon \end{pmatrix}$$

for $\epsilon < 1$. Irrespective of k, $\rho(E) = \epsilon$, while if $k = 0$, then the convergence factor in any reasonable norm should be ϵ. But if $k = 10^8$, let $\mathbf{e}_0 = [0, 1]^T$, and the convergence factor in the l^2-norm after one iteration is

$$\rho_1 = \frac{\|\mathbf{e}_1\|}{\|\mathbf{e}_0\|} = \|M\mathbf{e}_0\| = \sqrt{1000^2 + \epsilon^2} \approx 1000.$$

This demonstrates the downside of considering convergence for nonsymmetric AMG methods based on the spectral radius, namely, that iterations may diverge, and the spectral radius bound on the convergence factor is not necessarily achieved in practice. In the case of SPD A, the two-grid AMG error propagation operator, E_{TG}, is A-symmetric and $\|E_{TG}\|_A = \rho(E_{TG})$. In [18], energy-norm convergence is extended to nonsymmetric problems through $\sqrt{A^*A}$- and $\sqrt{AA^*}$-norms (using the square root to maintain the order of the problem). Here, for nonsingular A, energy minimization and RN AMG are related to two-grid theory in the $\sqrt{A^*A}$-norm.

Energy-minimization techniques for nonsymmetric problems have also been proposed in the A^*A-norm for P and in the AA^*-norm for R, as opposed to the A-norm as referenced in Lemma 4.2 (see, for example, fGMRES [41, (2.7)] and CGNR [41, (2.34)]), and are the basis of forming transfer operators in RN AMG for nonsymmetric problems. Building on Lemmas 4.1 and 4.2 gives the result in Lemma 4.6. Coupled with Conjecture 4.7, two-grid convergence follows from [18]. Lemma 4.6 provides a meaningful theoretical motivation for energy-minimization as applied to nonsymmetric problems.

LEMMA 4.6. *The solution to energy-minimization in the A^*A- and AA^*-norms satisfies the nonsymmetric strong approximation property in the $\sqrt{A^*A}$- and $\sqrt{AA^*}$-norms, respectively; that is, for all $\mathbf{v}$ there exists a $\mathbf{v}_{c_1}, \mathbf{v}_{c_2}$ such that*

$$\left\|\mathbf{v} - P_{\text{ideal}}^{A^*A}\mathbf{v}_{c_1}\right\|^2_{\sqrt{A^*A}} \leq \frac{K}{\|A^*A\|}\|\mathbf{v}\|^2_{A^*A},$$
$$\left\|\mathbf{v} - P_{\text{ideal}}^{AA^*}\mathbf{v}_{c_2}\right\|^2_{\sqrt{AA^*}} \leq \frac{K}{\|AA^*\|}\|\mathbf{v}\|^2_{AA^*}.$$

Proof. The proof follows immediately from the equivalence of the WAP(A^2) and SAP(A) (Lemma 4.1) and the convergence of energy-minimization to P_{ideal}, in this case P_{ideal} for A^*A and AA^* (Lemma 4.2). ∎

Conjecture 4.7 (stability). Let A be nonsingular. The nonorthogonal coarse-grid correction given by transfer operators $R = (P_{\text{ideal}}^{AA^*})^T$ and $P = P_{\text{ideal}}^{A^*A}$ is *stable*; that is,

$$\left\|P(RAP)^{-1}RA\right\|_{\sqrt{A^*A}} = \left\|I - P(RAP)^{-1}RA\right\|_{\sqrt{A^*A}} = C \tag{23}$$

for some constant C, independent of mesh spacing.

Remark 4.8. In the symmetric case, $R = P^T$ and $C = 1$, as the coarse-grid correction is an A-orthogonal projection. In the nonsymmetric case, the stability assumption necessary for two-grid convergence as shown in [18] is primarily to ensure a nonsingular and reasonably conditioned coarse-grid operator, RAP. Conjecture 4.7 appears to hold in general, but expanding the ideal operators and forming RAP or the full projection (23) does not provide a clear method to bound its norm. However, in practice the root-node approach over traditional aggregation offers greater stability of the nonorthogonal projection through enforcing the identity over C-points in transfer operators. Specifically, when forming the coarse-grid, the identity block in R and P help ensure the nonsingularity of RAP.

5. Numerical results. In the following, the open source software package PyAMG [6] is used to implement RN AMG[4] and SA AMG.[5] In each routine, cumulative estimates of the SC and CC (see section 2.4) are tracked in WUs. All costs in the setup phase are included, such as computing the triple matrix product RAP. In addition, the CC estimates include pre- and postsmoothing, computing and restricting the residual, and coarse-grid correction, but the coarsest-grid direct solve is not included. Measured CC multiplied by the time to perform one SpMV is highly accurate when compared with the wall-clock time of the solve phase. Although measured SC does not track as closely with wall-clock time (likely due to a more complicated setting with memory allocation, conditionals, etc.), it still provides a good estimate of setup cost and differs from wall-clock times by a small constant.

As with many AMG methods, RN AMG has several parameters. The optimal SOC measure is problem dependent, although the evolution measure tends to be more robust for problems with strong anisotropy. Still, the evolution measure does result in higher setup costs, as observed in section 5.1. Another parameter is that of the degree d of the sparsity pattern for P (see line 7 in Algorithm 3). Generally, more difficult problems such as anisotropic diffusion require d to be as large as 4–5. The number of smoothing iterations applied to P is generally set to $\lceil 1.5d \rceil$, as a larger sparsity pattern requires more iterations to minimize the energy of the columns. For SPD problems, the CG-based variant of energy minimization is the least costly, while GMRES is used for nonsymmetric problems (GMRES does not offer improved convergence for SPD problems; see Remark 5.1). Filtering P also requires user-level decisions (lines 9 and 17 in Algorithm 3). Based on experimentation, a drop tolerance of θ is most effective (in contrast to a fixed number of elements per row k) and is used in the following tests. Optimal values depend on the connectivity of the matrix and are beyond the scope of the numerical tests presented; however, $\theta \in [0.05, 0.25]$ is used in most scenarios.

In each setup phase below, a maximum coarse-grid size of 20 is used, and candidate vectors are improved (line 13, Algorithm 3) with four sweeps of relaxation. For cycling, each test uses an accelerated V-cycle to iterate to a relative residual tolerance of 10^{-8}. Details of the specific relaxation scheme, acceleration type, and other details are specified on a problem-by-problem basis below.

5.1. 3D-diffusion.

5.1.1. Filtering and complexity. The first example demonstrates the effectiveness of the proposed filtering strategy in reducing the CC and SC of an RN solver. Although filtering is a key component of RN AMG on nearly all problems, it is especially applicable in three dimensions, where there is high connectivity between nodes, resulting in relatively dense operators. Consider the anisotropic diffusion problem

$$u_{xx} + u_{yy} + 0.001u_{zz} = f. \tag{24}$$

Linear finite elements are used to discretize (24) on an unstructured tetrahedral mesh of the unit cube, with homogeneous Dirichlet boundaries, yielding a matrix with approximately 2.65M DOFs. While the anisotropy is aligned with the coordinate axis, the unstructured mesh yields a variety of non–grid-aligned anisotropies, known to be more difficult for AMG than the grid-aligned case (see section 5.2). Table 1 shows complexities and average convergence factors (ρ) for solving (24) using various

[4] See `root_node_solver`.

[5] See `smoothed_aggregation_solver`.

TABLE 1
Impact of filtering for the 3D-anisotropic diffusion problem.

Prefilter θ	–	0.1	0.2	–	–	0.1	0.2
Postfilter θ	–	–	–	0.1	0.2	0.1	0.2
SC	1157.9	161.7	123.2	581.5	432.9	156.2	129.8
OC	3.0	1.4	1.3	1.8	1.5	1.4	1.3
CC	18.9	8.0	7.1	10.1	8.0	7.7	6.9
ρ	0.52	0.60	0.62	0.51	0.53	0.59	0.61

TABLE 2
Break-down of setup cost in WUs for 3D-anisotropic diffusion.

Prefilter θ	Postfilter θ	Aggregation	Candidates	P	RAP	Total SC
–	–	478.7	24.1	469.7	271.1	1243.6
0.2	–	48.5	10.4	55.1	9.2	123.2
–	0.2	64.1	11.7	338.7	18.5	432.9
0.2	0.2	46.3	10.1	65.4	7.9	129.8

combinations of pre- and postfiltering on P. A V$(1,1)$-cycle with symmetric Gauss–Seidel relaxation and CG acceleration is used. The evolution SOC measure is used, with a drop tolerance of 4.0, and CG energy minimization with $d = 4$.

Table 1 shows up to an order of magnitude reduction in SC and more than a 50% reduction in CC by using filtering. While filtering does not guarantee a reduction in complexity, significant savings are often observed with only a marginal impact on convergence. In some situations, filtering has been observed to not only reduce cost, but also improve convergence (see section 5.4).

The SC in Table 1 is broken down into four main categories in Table 2. "Aggregation" is the cost of computing the strength matrix S and forming aggregates $\mathcal{A}$, most of which is due to using the evolution measure. "Candidates" refers to relaxing candidate set B and restricting B to a coarse level. Column "P" refers to the cost of forming the tentative interpolation operator and applying energy-minimization smoothing iterations to construct P, while column "RAP" represents a measure of the triple-matrix product. Each column gives the total cost for the given processes over all levels, measured in WUs.

Filtering P has a direct impact on all setup components that use R and P matrix operations (see "RAP" and "Candidates" in Table 2). Consequently, this reduces the cost of restricting the residual and the coarse-grid correction as measured in the CC, along with the cost of smoothing P, which is the focus of prefiltering. Table 2 also highlights the high cost of the strength measure in cases when coarse-grid complexity is not contained, i.e., filtering is not used, demonstrating the additional benefit of reduced costs on all subsequent grids through a sparser coarse-grid operator, $A_c = RAP$.

Remark 5.1. Although GMRES energy-minimization targets the SAP and CG the WAP, when applied to SPD problems, there is not a notable difference in convergence. For instance, if GMRES is used in Table 1, then the convergence rates change by no more than 0.005, and operator complexities remain essentially the same.

5.1.2. Modified evolution SOC. One contribution of this paper is the detailed SC estimates. Precise SC estimates are important when comparing AMG methods—e.g., components such as energy-minimization and the evolution SOC measure may

add significantly to the overall setup costs. Moreover, these estimates help identify areas that are opportune for cost reduction.

In considering Table 2, one possible area for cost reduction is the aggregation phase, where the cost of the SOC computation dominates. Traditional SOC measures require only a few work units, usually 2 or 3 times the operator complexity. However, Table 2 shows that evolution-based SOC is a substantial part of the setup phase, which is attributed to the global spectral radius estimate used in the weighted Jacobi method [39]. This estimate is calculated with an Arnoldi/Lanczos process and costs roughly 15 matrix-vector multiplies on each level. However, alternative methods [4] use ℓ_1-Jacobi relaxation to provide an inexpensive local rowwise weight. This alternative is explored here for the evolution SOC measure.

Table 3 depicts detailed SC results for using the modified ℓ_1-Jacobi evolution measure (cf. Table 2). This change results in similar operator and cycle complexities, and nearly identical convergence rates, as for the original evolution measure. When comparing Tables 2 and 3, it is apparent that the aggregation phase has been significantly reduced in cost. Additionally, when examining the most efficient solvers, where prefiltering uses $\theta = 0.2$, the overall savings in the setup complexity are roughly 20%.

TABLE 3
Break-down of setup cost in WUs for 3D-anisotropic diffusion.

Prefilter θ	Postfilter θ	Aggregation	Candidates	P	RAP	Total SC
–	–	449.1	24.4	473.0	249.5	1195.9
0.2	–	25.5	10.4	53.6	7.6	97.1
–	0.2	43.2	11.7	326.7	16.9	398.6
0.2	0.2	22.7	10.2	63.3	6.9	103.1

5.2. Totally anisotropic diffusion. Diffusion-like operators are prototypical AMG problems, as they are elliptic and SPD. However, with strong anisotropy, these problems still pose a challenge to multilevel solvers. A 3D example is used in section 5.1 to demonstrate filtering. In this example, consider a 2D rotated anisotropic diffusion problem on the domain Ω of the form

$$\begin{aligned} -\nabla \cdot Q^T D Q \nabla u &= f \quad \text{for } \Omega = [0,1]^2, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

where

$$Q = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}, \qquad D = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix}.$$

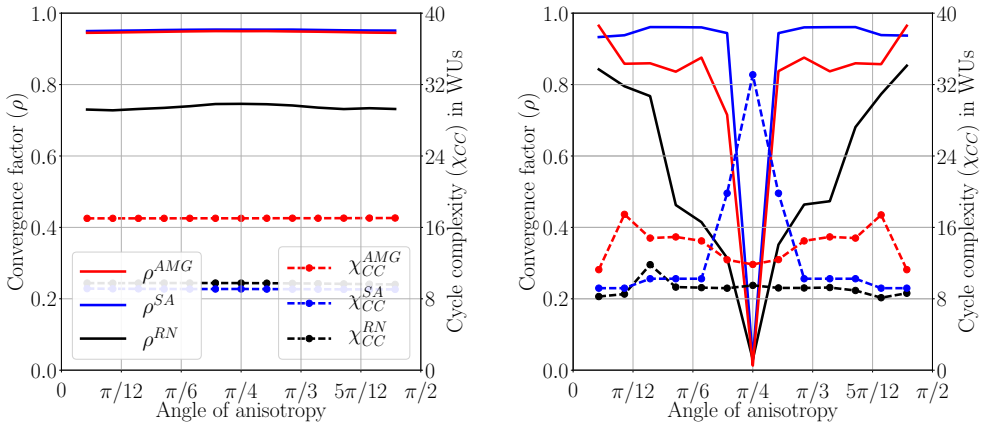
Here, ϵ represents anisotropy, and ψ the angle of rotation from the coordinate axis, both of which contribute to the difficulty of this problem [9, 10, 12, 13, 15, 21, 22, 23, 27, 33, 36, 39, 41, 45]. To see this, consider the nonrotated case of $\psi = 0$, which has a spectrum of the form

$$\begin{aligned} -\nabla \cdot D \nabla u_{jk} &= \lambda_{jk} u_{jk}, \quad \text{where} \\ u_{jk} &= \sin(j\pi x)\sin(k\pi y), \\ \lambda_{jk} &= \pi^2(j^2 + \epsilon k^2), \end{aligned}$$

for $j, k \in \mathbb{Z}^+$. If $\epsilon = 1$, then the lowest energy mode is the lowest Fourier mode, $u_{11} = \sin(\pi x)\sin(\pi y)$, which is locally representative of all low-energy modes. However, if

$\epsilon \approx 0$, for small j there are high-frequency eigenfunctions in the y-direction ($k \gg 0$) with relatively small eigenvalues that are no longer represented locally by the lowest Fourier mode. As a result, relaxation schemes are unable to capture these modes, while coarse-grid correction is not equipped to handle such *hidden* high-frequency error.

For angles ψ aligned with the mesh, line relaxation or semicoarsening along the direction of anisotropy can be used [44]. However, for strong anisotropies, $\epsilon \approx 0$, with angles that are not aligned with the mesh, efficient and effective multigrid solvers remain elusive. This section considers a finite element discretization of strongly and *totally* anisotropic diffusion, $\epsilon = 0.001$ and $\epsilon = 0$, respectively, on a unit square with Dirichlet boundary conditions. Totally anisotropic diffusion is particularly challenging as the problem is effectively reduced to a sequence of 1D problems on a 2D domain.



(a) Unstructured mesh: $n = 5 \times 10^6$.

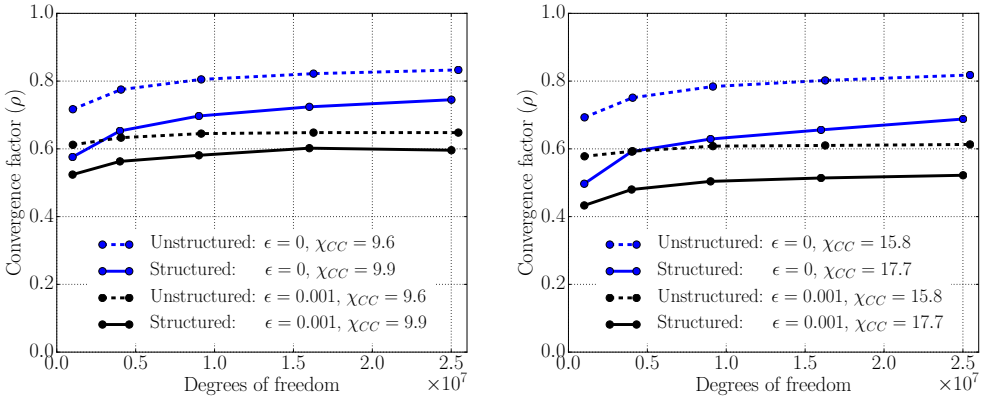
(b) Structured mesh: $n = 8 \times 10^6$.

FIG. 6. *Results for totally anisotropic diffusion ($\epsilon = 0$), with angles in $(0, \pi/2)$. A structured grid of size 2000×2000 and an unstructured mesh with resolution $h \approx 1/2000$ are used. Legend entries in* (a) *apply to* (b) *as well. RN AMG, shown in black, outperforms SA AMG and CF AMG in all cases.*

Figure 6 compares RN AMG, CF AMG, and SA AMG applied to totally anisotropic diffusion for various angles. All solves use a symmetric V(1,1)-cycle with symmetric Gauss–Seidel relaxation and CG acceleration. Grid-aligned anisotropies are generally easier to solve than non–grid-aligned; hence the excellent convergence factors at $\theta = \pi/4$ on the structured mesh. In the case of an unstructured mesh, all angles are effectively non–grid-aligned, resulting in consistent performance across angles.

CF AMG uses a classical strength measure with drop tolerance of 0.5 and standard CF-splitting and interpolation [42]. Smoothed aggregation uses a symmetric strength measure (with a drop tolerance of 0.0—i.e., the strength matrix is given by A and normalized as in (7)) and two iterations of weighted Jacobi interpolation smoothing [48]. The root-node solver in this case uses two steps of an evolution strength measure (with a drop tolerance of 4.0), along with six iterations of CG energy-minimization smoothing of P. For the energy minimization, a degree $d = 4$ sparsity pattern is used with filtering, $\theta_{\text{pre}} = \theta_{\text{post}} = 0.1$. Two and three Jacobi smoothing iterations were applied to the SA AMG solver in an attempt to mimic the expanded sparsity pattern used in RN AMG, but the convergence with respect to CC did not improve.

Figure 6 highlights the effectiveness of RN AMG over all angles with moderate operator and cycle complexities. For time-to-solution with respect to floating point operations and wall-clock time, RN AMG achieves 3–30× speed-up in comparison to SA AMG and CF AMG on a structured or unstructured mesh, with moderate cycle complexities in all cases. It should be noted that performance of SA AMG improves when using the modified evolution SOC measure introduced in section 5.1.2, but RN AMG still performs 2–3× faster with respect to time and complexity. This is a notable achievement in performance for this problem, as anisotropic diffusion remains a significant challenge to most solvers.



(a) V(1, 1)-cycle. (b) W(1, 1)-cycle.

FIG. 7. *Scaling results for RN AMG for anisotropic diffusion with* $\epsilon = 0.001$, $\epsilon = 0$, *and* $\theta = 3\pi/16$. *Cycle complexity,* χ_{CC}, *is shown and is constant for all problem sizes.*

Figure 7 demonstrates the scaling of convergence factors as problem size increases for $V(1,1)$- and $W(1,1)$-cycles. In the case of $\epsilon = 0.001$, V-cycle convergence factors asymptote and scale perfectly, independent of h, on structured and unstructured meshes, up to 25 million unknowns; the SC and CC also scale but are not shown. However, in the case of $\epsilon = 0$ there is a slow growth in the convergence factor as the problem size increases for both V- and W-cycles. To analyze, consider a convergence factor, $\rho(h)$, dependent on spatial step size h:

$$\rho = \bar{\rho}(1 - ah^q), \tag{25}$$

where $q = 1$ in the case of linear finite elements, a is some constant, and $\bar{\rho}$ is the asymptotic convergence factor, $\lim_{h\to 0} \rho = \bar{\rho}$. A log of (25) and an expansion yields $\log(1 - ah) = -ah + O(h^2)$ or $-\log(\rho) = -\log(c) + ah$. A linear fit on the three smallest step sizes in Figure 7 for $\epsilon = 0$ leads to the following asymptotic convergence factors:

$$\text{structured:} \quad \rho \to 0.82, \qquad \text{unstructured:} \quad \rho \to 0.88.$$

Although an asymptotic convergence factor of approximately 0.88 is relatively slow, *scalable* convergence of totally anisotropic diffusion on unstructured meshes has not been achieved by other AMG methods.

In practice, using W-cycles over V-cycles should increase the accuracy of the coarse-grid correction. However, Figure 7 reveals that W-cycles offer only minor improvements. This indicates that the algebraically smooth error is not well represented

by the coarse-grid, and that improved strength measures and coarsening routines [9, 12, 13, 22, 36, 39] may improve the coarse-grid, thereby improving convergence for this problem.

5.3. Recirculating flow (nonsymmetric). One of the benefits of the root-node approach is the ability to handle a variety of problems, including nonsymmetric problems and systems, without redesign of methodology and implementation. In this section, a standard recirculating flow example known as the *double-glazing problem* is used [24], which models temperature distribution over a domain when an external wall is hot. The governing PDE is given by

$$-\epsilon\nabla\cdot\nabla u+\mathbf{b}(\mathbf{x})\cdot\nabla u=f, \tag{26}$$

where $\epsilon=0.005$, and wind is given by $\mathbf{b}(\mathbf{x})=[2x_1(1-x_0^2),-2x_0(1-x_1^2)]$. Dirichlet boundaries are imposed on the domain $[0,1]\times[0,1]$, where $u=0$ on the north, south, and west sides of the domain, and $u=1$ on the east, leading to boundary layers near corners with discontinuities.

A standard Galerkin finite element method (GFEM) based on a regular triangular mesh is used, resulting in a nonsymmetric discrete linear system. Multigrid theory for nonsymmetric problems is less developed in comparison to the symmetric case; however, SA AMG has been extended to nonsymmetric problems [43] and is used in this section (see Algorithm 3).

Each example uses a V(1, 1)-cycle of weighted-Jacobi relaxation with GMRES acceleration. While Jacobi relaxation is not guaranteed to converge for a nonsymmetric problem, it remains around half the cost of using relaxation on the normal equations. In section 5.4 an upwind discretization is considered which requires relaxation on the normal equations for effective convergence. For SA AMG, a classical strength measure (with drop tolerance 0.25) is used along with one and three steps of Jacobi smoothing applied to P, labeled SA_1 and SA_3, respectively. The symmetric traditional SA strength measure is not used, due to the nonsymmetry of the problem. Two steps of the evolution measure (with drop tolerance 3.0) is used for RN AMG along with two iterations of GMRES energy minimization for P with $d=1$ (labeled RN_1), and five iterations of GMRES energy minimization for P with $d=3$ (labeled RN_3).

TABLE 4
Nonsymmetric SA AMG and RN AMG for the recirculating flow problem.

		2000 × 2000				3000 × 3000				4000 × 4000			
θ	d	SC	OC	CC	ρ	SC	OC	CC	ρ	SC	OC	CC	ρ
–	SA_1	72	1.4	5.2	0.74	71	1.4	5.2	0.82	71	1.4	5.2	0.88
	SA_3	229	2.3	11.2	0.96	230	2.3	11.2	0.96	227	2.3	11.2	0.93
–	RN_1	98	1.4	5.1	0.46	96	1.4	5.0	0.50	95	1.4	4.9	0.45
	RN_3	403	2.5	9.7	0.68	407	2.4	9.4	0.63	405	2.3	9.2	0.76
0.1	RN_1	126	1.4	5.1	0.52	125	1.4	5.0	0.56	124	1.4	4.9	0.56
	RN_3	284	1.8	6.7	0.53	287	1.8	6.7	0.52	285	1.8	6.6	0.63

Table 4 demonstrates that optimal results are achieved for this example with no filtering and a small sparsity pattern ($d=1$). This agrees with practical experience: generally it is effective to increase the filtering tolerance as the degree of the sparsity pattern for P increases, or as the connectivity of matrix A increases. This is observed in the 3D-anisotropic diffusion problem, where high connectivity and a $d=4$ sparsity pattern allow for a large $\theta=0.2$. If the sparsity pattern increases in distance from

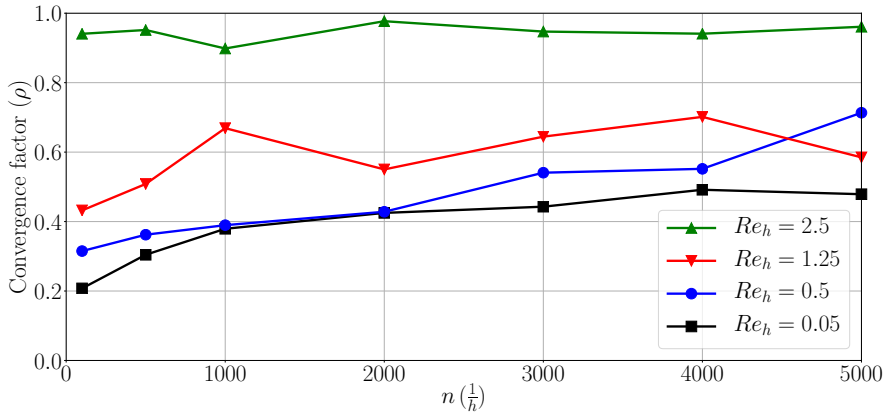


FIG. 8. *RN AMG convergence factor as a function of h for fixed half-grid Reynolds numbers,* $R_h \in \{0.05, 0.5, 1.25, 2.5\}$.

the root-node, or if the matrix is highly connected, then it is likely there are entries that are not critical to performance and are candidates for removal.

Classical CF AMG is not designed for nonsymmetric problems, making RN AMG the clear choice for a problem such as this. RN AMG achieves more than a 6× speed-up over SA AMG for the largest problem size considered, *with a lower CC* and only slightly larger SC. Furthermore, RN AMG convergence factors with degree $d = 1$ appear to have reached an asymptote, while SA AMG is still demonstrating a steady increase with problem size.

Remark 5.2. With the recirculating flow, as the grid-size (h) approaches zero, there are two competing factors that contribute to the numerical difficulty of the problem. As $h \to 0$, the diffusive part of the problem, $-\epsilon\nabla \cdot \nabla$, becomes increasingly dominant, as the discretization scales like $\frac{1}{h^2}$ while $\mathbf{b}(\mathbf{x})\cdot\nabla$ scales like $\frac{1}{h}$. The resulting linear system is more symmetric and diffusion-like, which is preferable for AMG. However, as $h \to 0$, convergence factors often increase to an asymptotic value (see (25) and Figures 7(a) and 7(b)), due to smaller eigenvalues and an increasing number of levels in the AMG hierarchy. Together, these factors correspond to the so-called *half-grid Reynolds number* or *cell Reynolds number*, $R_h = \frac{|\mathbf{b}|h}{2\epsilon}$, where convergence factors are expected to be consistent for a fixed R_h and degrade for $R_h \gg 1$. Figure 8 demonstrates this phenomenon. Convergence factors asymptote for each half-grid Reynolds number as $h \to 0$. In this case, convergence degrades by a factor of 10 when increasing from $R_h = 1.25$ to $R_h = 2.5$; thus it is faster to solve a refined problem with $R_h = 1.25$ (and several times as many DOFs), rather than a system with $R_h = 2.5$.

Based on the scalability of RN AMG performance for fixed R_h, with the appropriate grid-size, most convection-diffusion problems as in (26) should exhibit scalability in convergence. Of course, this relies on the diffusion operator dominating the discrete system. As a limiting case, RN AMG is applied to the steady-state transport equation in the following section.

5.4. Upwind transport (nonsymmetric). In this section, a highly nonsymmetric upwind discretization of the steady-state transport equation is considered. Define the domain as $\Omega = (0,1) \times (0,1)$, with constrained inflow boundaries $\Gamma_{\text{in}} = \{(x,y) : x = 0 \text{ or } y = 0\}$ and free outflow boundaries $\Gamma_{\text{out}} = \{(x,y) : x = 1 \text{ or } y =$

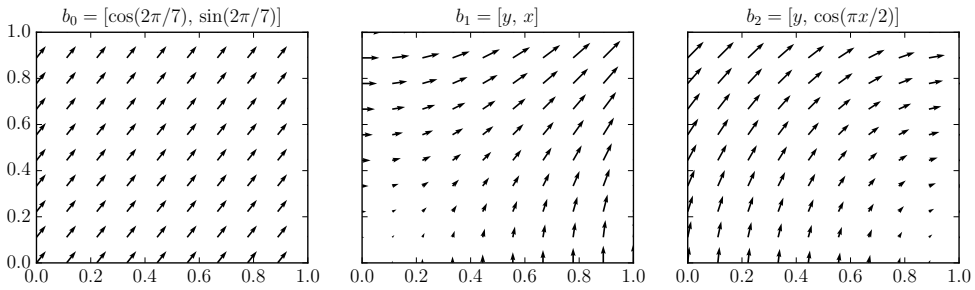


FIG. 9. *Vector fields of flow functions,* $\mathbf{b}(x, y)$ *considered in* (27)*, with inflow boundaries* $\Gamma_{in} = \{(x, y) : x = 0 \text{ or } y = 0\}$.

1}. The steady-state transport equation is then given as

$$\begin{aligned} \mathbf{b}(x,y)\cdot\nabla u + c(x,y)u &= f(x,y) \quad \Omega, \\ u &= g(x,y) \quad \Gamma_{\text{in}}. \end{aligned} \tag{27}$$

RN AMG performance is demonstrated for variations in $\mathbf{b}(x, y)$ and $c(x, y)$, using an upwind lumped bilinear discontinuous (LBLD) finite element discretization [34]. A discontinuous discretization is used to allow for discontinuities in material coefficient $c(x, y)$, and to account for the fronts that develop in hyperbolic-type PDEs. The transport problem, (27), results in a highly nonsymmetric (nearly lower-triangular in the proper ordering) matrix A, for which traditional relaxation schemes such as Jacobi and Gauss–Seidel relaxations diverge, even for small problems. Thus two sweeps of Gauss–Seidel relaxation on the normal equations are chosen as the relaxation scheme, which contributes to a larger CC.

One anomalous feature of the LBLD discretization applied to (27) is that the best convergence rates are obtained with aggressive filtering, $\theta_{pre} = \theta_{post} = 0.4 - 0.5$. The rationale is that a large degree sparsity pattern with aggressive filtering results in a sparsity pattern that is long and narrow, following the direction of flow. For a hyperbolic PDE with the solution following characteristic curves, effective sparsity patterns for a given column of P will align with the characteristic associated with the root-node. However, in practice this requires either a priori knowledge of the problem, to motivate aggressive filtering, or experimentation. For most problems, filtering values of $\theta_{pre} = \theta_{post} \in [0.05, 0.15]$ are effective choices, increasing to $\theta_{pre} = \theta_{post} \in [0.1, 0.25]$ for problems with high connectivity—e.g., 3D-diffusion, as discussed in section 5.1. However, the LBLD results demonstrate that further work is needed to algebraically determine the optimal interpolation sparsity pattern for a given problem.

RN AMG is applied to the LBLD discretization of (27) for three different material coefficients $c(x, y)$, as shown in (28), and three directional functions, as shown in Figure 9:

$$\begin{aligned} \text{Constant:} \quad & c_1(x,y) = 1, \\ \text{Square in square (SnS):} \quad & c_2(x,y) = \begin{cases} 10^4, & x,y \in [0.25, 0.75], \\ 10^{-4}, & x,y \notin [0.25, 0.75], \end{cases} \\ \text{Split:} \quad & c_3(x,y) = \begin{cases} 10^{-4}, & x < 0.5, \\ 10^4, & x \geq 0.5. \end{cases} \end{aligned} \tag{28}$$

TABLE 5
Nonsymmetric RN AMG for variations on the 2D steady-state transport equation (27). Matrices have 4,000,000 DOFs and 15,986,004 nonzeros.

	$\mathbf{b}(x,y)$	$[\cos(2\pi/7), \sin(2\pi/7)]$			$[y, x]$			$[y, \cos(\pi x/2)]$		
	$c(x,y)$	1	SnS	Split	1	SnS	Split	1	SnS	Split
RN	SC	164	147	130	125	105	107	126	112	109
	OC	1.88	1.76	1.66	1.92	1.83	1.78	1.91	1.83	1.77
	CC	17.9	16.7	15.8	18.2	17.4	17.0	18.2	17.4	16.8
	ρ	0.79	0.74	0.75	0.86	0.84	0.85	0.87	0.87	0.86

Results are shown in Table 5 for a discretization with 4,000,000 DOFs, sufficiently large to observe asymptotic convergence factors. A classical SOC with drop tolerance $\theta = 0.35$ is used, and GMRES energy-minimization is applied to a degree four sparsity pattern for interpolation and restriction operators in RN AMG, with filtering $\theta_{pre} = \theta_{post} = 0.45$. The multigrid solver is used as a preconditioner for GMRES, and problems are solved to 10^{-8} residual tolerance. Candidate vectors are taken as the constant and are *not* improved for this problem, as convergence tended to degrade.

Although convergence factors in the case of nonconstant flow are higher than desired (worst case $\rho \approx 0.87$ for RN AMG), it is encouraging that AMG methods are able to solve upwind discretizations of a hyperbolic PDE with discontinuous and nonconstant coefficient functions. Convergence factors of SA AMG quickly approached one, and did not converge in 500 iterations.

Often when considering transport-type problems, "sweeps" are performed, where the problem is discretized in angle and a linear solve performed for each angle. Transport sweeps are an important part of the DSA algorithm for models of neutral particle transport [2]. For this reason, it is of interest to demonstrate RN AMG's capability on the entire spectrum of angles, similar to anisotropic diffusion as considered in section 5.2. Figure 10 shows convergence factors of RN AMG as a function of direction of flow, $\theta \in [0, \pi/2]$, as applied to the LBLD discretization of (27).

5.5. Elasticity example (systems problem). AMG for systems of PDEs is an important topic due to the vast number of problems formulated this way. Although there are some problem-specific extensions to CF AMG for systems such as elasticity [5], where rigid body modes are incorporated into a modified method, CF AMG is not applicable to systems in a general setting. In contrast, SA AMG specifically targets systems-based problems such as elasticity and performs well in many cases. Here, the suitability of RN AMG for systems is illustrated through a comparison with SA AMG for a standard elasticity model problem, a problem for which SA AMG was designed.

The test problem is isotropic linearized elasticity defined by

$$-\operatorname{div}\left(\lambda \operatorname{tr}\left(\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)/2\right) I + \mu\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)\right) = f, \tag{29}$$

where λ and μ are the Lamé parameters, I is the identity matrix, and tr() is the trace function. Example two from the MFEM package [1] is used to discretize (29) using a regular mesh of a beam that is eight times longer than it is wide (`data/beam-tri.mesh`). Three material choices, of increasing difficulty for AMG, are considered: a steel beam, with Young's modulus and Poisson ratio of $E = 180\text{E}9$ and $\nu = 0.30$; a rubber beam, with $E = 0.1\text{E}9$ and $\nu = 0.499$; and a more difficult rubber beam, where $\nu = 0.4999$. In general, the closer ν is to 0.5, the more difficult

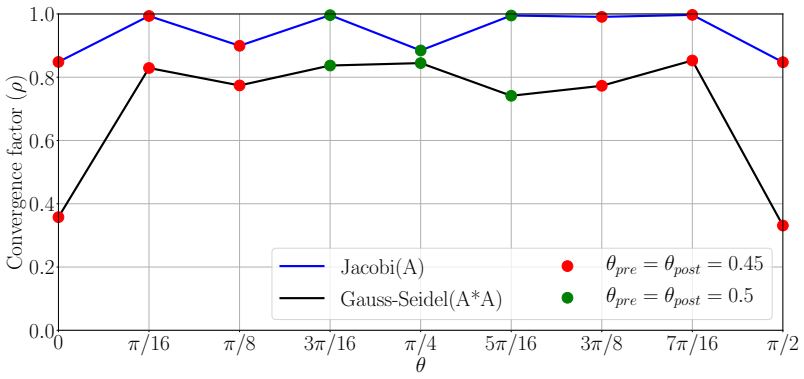


FIG. 10. *Convergence factors for GMRES-accelerated RN AMG applied to the steady-state transport equation, as a function of the direction of flow* $\mathbf{b}(x, y) = (\cos(\theta), \sin(\theta))$. *Results are shown for the Gauss–Seidel on the normal equations (GSNE) relaxation as well as Jacobi relaxation, to demonstrate why normal-equation relaxation is necessary for highly nonsymmetric problems. Other solver parameters are fixed for all tests, with the exception of filtering, which is modified slightly for improved convergence on angles close to* $\pi/4$. *Convergence with GSNE is in the range* $[0.75, 0.85]$ *for interior angles, with significant improvement close to the boundaries.*

the system is for AMG to solve. The east side of the beam is fixed to a wall with $\mathbf{u} = \mathbf{0}$ Dirichlet boundary conditions. Neumann conditions are applied on other boundaries, with $\mathbf{u} \cdot \mathbf{n} = f$, where $f = 0$ for the north and south sides and $f = -0.01$ on the west side, representing a downward force. Standard linear triangular finite elements are used to produce an SPD system with two variables at each spatial location and the corresponding block sparse row matrix with 2×2 blocks.

Each example uses a V$(1, 1)$-cycle of symmetric block Gauss–Seidel relaxation with CG acceleration. Both RN AMG and SA AMG use the classical strength measure with a drop tolerance of 0.5. RN AMG does not use the evolution measure here because the classic measure performed well and lowered the setup complexity.[6] The terms SA_k and RN_k again refer to the degree of interpolation smoothing. For RN_4, six iterations of CG energy minimization are used, while for RN_1, two iterations are used. Experimentation indicates that only prefiltering is needed, so θ values refer only to prefiltering. The candidate vectors B_0 are the three rigid-body-modes in two dimensions, two translations, and one rotation.

Table 6 gives the numerical results for three problem sizes, from 256k to 4M DOFs. Various filtering values produce qualitatively similar results, so a single result is reported. All methods converge well for the steel beam, with the 0.5 convergence rate for RN_1 being offset by the low operator complexity of 1.2. The rubber beams are considerably harder, with the most difficult beam problem yielding convergence rates well above 0.9.

Comparing the methods with each other, the results reveal that RN_1 converges

[6] If the three elasticity test cases are rerun for RN_4 using the evolution strength measure, then the convergence rate degrades by about 0.01–0.02, and the operator complexities remain similar; i.e., RN AMG using the evolution measure converges similarly to SA using the classic measure. While SOC for systems is an active topic of AMG research and not well understood, one possible reason for this small difference between the two measures is that the "anisotropy" in the beam is grid-aligned, and the classical measure is known to handle grid-aligned anisotropies well.

TABLE 6

Application of SA AMG and RN AMG to three linearized elasticity problems. The tests are progressively more difficult, with Test 1 *being a steel bar with Poisson ratio* $\nu = 0.30$*, Test* 2 *being a rubber bar with* $\nu = 0.499$*, and Test* 3 *being a rubber bar with* $\nu = 0.4999$*.*

	Grid size:		1024×128				2048×256				4096×512			
Test	θ	d	SC	OC	CC	ρ	SC	OC	CC	ρ	SC	OC	CC	ρ
1	–	SA_1	105	1.6	9.8	0.23	106	1.7	9.8	0.26	105	1.7	9.8	0.26
	0.1	RN_1	84	1.2	6.9	0.50	84	1.2	6.9	0.49	85	1.2	6.9	0.51
		RN_4	226	1.4	8.2	0.19	226	1.4	8.2	0.21	225	1.4	8.2	0.19
2	–	SA_1	104	1.6	9.6	0.83	106	1.6	9.8	0.84	104	1.7	9.8	0.84
	0.1	RN_1	77	1.2	6.7	0.92	77	1.2	6.7	0.94	78	1.2	6.7	0.93
		RN_4	221	1.4	8.2	0.80	223	1.4	8.2	0.80	223	1.4	8.3	0.81
3	–	SA_1	104	1.6	9.7	0.93	105	1.6	9.8	0.94	105	1.7	9.8	0.93
	0.1	RN_1	82	1.2	6.7	0.97	82	1.2	6.7	0.97	81	1.2	6.7	0.97
		RN_4	228	1.4	8.2	0.92	229	1.4	8.3	0.92	229	1.4	8.3	0.92

slowly in comparison to SA_1 or RN_4. This is because the three candidate vectors provided as interpolation constraints require a richer sparsity pattern than $d = 1$ in order to satisfy the constraints.

Another important point is that RN AMG with $d = 4$ achieves a better convergence rate and smaller operator complexity than SA AMG but at the cost of a larger SC. Taking the largest problem size for Test 2, with 4M DOFs (a grid size of 4096×512), the difference in convergence rate yields 87 iterations for RN_4 versus 107 iterations for SA_1. However, even with this faster convergence, SA AMG uses the fewest overall WUs.[7] SA AMG similarly provides slightly better overall work units for setup and solve with Test 3. In conclusion, this section shows that RN AMG is a viable approach to solving systems and is competitive with SA AMG for the considered elasticity problem, for which SA AMG was originally designed. Applying RN AMG to systems is still an active research topic, focusing on issues such as reducing the setup cost and considering more complicated examples.

Remark 5.3. It is important to note that the ideal P in (20) is ideal only for the specific two-grid setting outlined in [25]. This ideal P is not guaranteed to be ideal for multilevel methods, or even for all two-grid methods. However, the ideal P is still a useful target for interpolation, as evidenced by the experiments in this work and by the fact this it is ideal under the assumptions in [25]. Along with approximating P_{ideal}, RN AMG combines other proven approaches to constructing quality interpolation operators, interpolation smoothing, and guaranteed interpolation of known algebraically smooth modes.

As an example of how this combination of strategies makes RN AMG more robust than using P_{ideal}, two-grid experiments for the three test cases from Table 6 are run on a small mesh yielding 4.5k DOFs. For the steel beam (Test 1), the convergence factors are 0.19, 0.17, and 0.18, for SA_1, RN_4, and ideal P, respectively. Moving to the rubber beam (Test 2), the convergence factors are 0.80, 0.79, and 0.97. Finally, for the harder rubber beam (Test 3), the convergence factors are 0.92, 0.91, and 0.98. Experimentally, it appears that as the Poisson ratio approaches 0.5, convergence P_{ideal} degrades significantly faster than the interpolation operators produced by SA AMG and RN AMG.

[7] It is possible, given the faster creep in convergence rate suffered by SA AMG, that RN AMG could outperform overall as the problem size increases.

6. Conclusions. The presented RN AMG methodology has proven successful on a wide variety of problems. The proposed filtering of the interpolation sparsity patterns before and after energy-minimization smoothing iterations greatly reduces setup and cycle complexity, and at times improves convergence, making RN AMG a robust solver for difficult SPD problems, nonsymmetric problems, and systems-related problems. Total complexity remains reasonable for all of the examples tested, and tends to scale with the size of the problem. Providing such total complexity estimates, including the setup phase, is a specific contribution of this work.

One particularly difficult problem is strongly anisotropic diffusion, which RN AMG is able to solve effectively, in contrast to other AMG methods. Because the focus here is on a general root-node methodology, coupled with energy-minimization smoothing of P, minimal testing is done in coupling RN AMG with other advancements in AMG. In the case of strongly anisotropic diffusion, many other works consider modified SOC and advanced coarsening schemes in order to better capture the anisotropy [9, 12, 13, 22, 36, 39], thereby improving convergence rates. Future work involves coupling RN AMG with a larger variety of SOC measures, aggregation routines, and sparsity patterns for P, to further improve convergence on anisotropic problems. Root-node is also shown to be effective at solving convection-diffusion and a discontinuous upwind discretization of a hyperbolic PDE (steady-state transport), demonstrating that AMG need not be limited to elliptic problems.

Parallelization of RN AMG is straightforward. The basic computational kernels are typically available in AMG codes (e.g., matrix-matrix multiply), are light-weight (e.g., filtering entries), or purely local in computation (e.g., the rowwise projection operation (8)).

New theoretical motivation for coupling energy minimization with RN AMG is provided in the symmetric and nonsymmetric setting. Proving Conjecture 4.7 would complete the two-grid convergence proof for ideal operators in the nonsymmetric case. However, the stability constraint is of limited use in practice, as it is not directly approximated like the weak and strong approximation properties. Further work on developing energy-based, nonsymmetric convergence theory is important in understanding how to construct AMG solvers for nonsymmetric systems.

REFERENCES

[1] *MFEM: Modular Finite Element Methods*, free library, https://mfem.org.

[2] M. L. Adams and E. W. Larsen, *Fast iterative methods for discrete-ordinates particle transport calculations*, Progr. Nuclear Energy, 40 (2002), pp. 3–159.

[3] A. H. Baker, R. D. Falgout, T. Gamblin, T. V. Kolev, M. Schulz, and U. M. Yang, *Scaling algebraic multigrid solvers: On the road to exascale,* in Competence in High Performance Computing 2010: Proceedings of an International Conference on Competence in High Performance Computing, Springer, Berlin, Heidelberg, 2012, pp. 215–226, https://doi.org/10.1007/978-3-642-24025-6_18.

[4] A. H. Baker, R. D. Falgout, T. V. Kolev, and U. M. Yang, *Multigrid smoothers for ultraparallel computing*, SIAM J. Sci. Comput., 33 (2011), pp. 2864–2887, https://doi.org/10.1137/100798806.

[5] A. H. Baker, T. V. Kolev, and U. M. Yang, *Improving algebraic multigrid interpolation operators for linear elasticity problems*, Numer. Linear Algebra Appl., 17 (2010), pp. 495–517, https://doi.org/10.1002/nla.688.

[6] W. N. Bell, L. N. Olson, and J. B. Schroder, *PyAMG: Algebraic multigrid solvers in Python*, v3.0, 2015, https://github.com/pyamg/pyamg.

[7] A. Brandt, *General highly accurate algebraic coarsening*, Electron. Trans. Numer. Anal., 10 (2000), pp. 1–20.

[8] A. Brandt, J. Brannick, K. Kahl, and I. Livshits, *Bootstrap AMG*, SIAM J. Sci. Comput., 33 (2011), pp. 612–632, https://doi.org/10.1137/090752973.

[9] A. Brandt, J. Brannick, K. Kahl, and I. Livshits, *Algebraic distance for anisotropic diffusion problems: Multilevel results*, Electron. Trans. Numer. Anal., 44 (2015), pp. 472–496.

[10] A. Brandt, J. J. Brannick, K. Kahl, and I. Livshits, *An Algebraic Distances Measure of AMG Strength of Connection*, preprint, 2011.

[11] 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.

[12] J. Brannick, M. Brezina, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *An energy-based AMG coarsening strategy*, Numer. Linear Algebra Appl., 13 (2006), pp. 133–148, https://doi.org/10.1002/nla.480.

[13] J. Brannick, Y. Chen, and L. Zikatanov, *An algebraic multilevel method for anisotropic elliptic equations based on subgraph matching*, Numer. Linear Algebra Appl., 19 (2012), pp. 279–295, https://doi.org/10.1002/nla.1804.

[14] J. Brannick and L. Zikatanov, *Algebraic multigrid methods based on compatible relaxation and energy minimization*, in Domain Decomposition Methods in Science and Engineering, Lecture Notes Comput. Sci. Engrg., Springer, Berlin, Heidelberg, 2007, pp. 15–26, https://doi.org/10.1007/978-3-540-34469-8_2.

[15] J. J. Brannick and R. D. Falgout, *Compatible relaxation and coarsening in algebraic multigrid*, SIAM J. Sci. Comput., 32 (2010), pp. 1393–1416, https://doi.org/10.1137/090772216.

[16] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation (αSA) multigrid*, SIAM Rev., 47 (2005), pp. 317–346, https://doi.org/10.1137/050626272.

[17] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive algebraic multigrid*, SIAM J. Sci. Comput., 27 (2006), pp. 1261–1286, https://doi.org/10.1137/040614402.

[18] M. Brezina, T. Manteuffel, S. McCormick, J. Ruge, and G. Sanders, *Towards adaptive smoothed aggregation (αSA) for nonsymmetric problems*, SIAM J. Sci. Comput., 32 (2010), pp. 14–39, https://doi.org/10.1137/080727336.

[19] M. Brezina, P. Vanek, and P. S. Vassilevski, *An improved convergence analysis of smoothed aggregation algebraic multigrid*, Numer. Linear Algebra Appl. 19 (2012), pp. 441–469, https://doi.org/10.1002/nla.775.

[20] T. F. Chan and P. Vanek, *Detection of strong coupling in algebraic multigrid solvers*, in Lecture Notes in Computational Science and Engineering, Springer, Berlin, Heidelberg, 2000, pp. 11–23, https://doi.org/10.1007/978-3-642-58312-4_2.

[21] M.-H. Chen and A. Greenbaum, *Analysis of an aggregation-based algebraic two-grid method for a rotated anisotropic diffusion problem*, Numer. Linear Algebra Appl., 22 (2015), pp. 681–701, https://doi.org/10.1002/nla.1980.

[22] P. D'Ambra and P. S. Vassilevski, *Adaptive AMG with coarsening based on compatible weighted matching*, Comput. Visualization Sci., 16 (2013), pp. 59–76, https://doi.org/10.1007/s00791-014-0224-9.

[23] 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, https://doi.org/10.1002/nla.559.

[24] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Numerical Mathematics & Scientific Computation, Oxford, UK, 2014.

[25] R. D. Falgout and P. S. Vassilevski, *On generalizing the algebraic multigrid framework*, SIAM J. Numer. Anal., 42 (2004), pp. 1669–1693, https://doi.org/10.1137/S0036142903429742.

[26] R. D. Falgout and U. M. Yang, *hypre: A Library of High Performance Preconditioners*, Springer, Berlin, Heidelberg, 2002, pp. 632–641, https://doi.org/10.1007/3-540-47789-6_66.

[27] M. W. Gee, J. J. Hu, and R. S. Tuminaro, *A new smoothed aggregation multigrid method for anisotropic problems*, Numer. Linear Algebra Appl., 16 (2009), pp. 19–37, https://doi.org/10.1002/nla.593.

[28] V. E. Henson and U. M. Yang, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177, https://doi.org/10.1016/S0168-9274(01)00115-5.

[29] X. Hu, P. S. Vassilevski, and J. Xu, *A two-grid SA-AMG convergence bound that improves when increasing the polynomial degree*, Numer. Linear Algebra Appl., 23 (2016), pp. 746–771, https://doi.org/10.1002/nla2053.

[30] C. Ketelsen, T. Manteuffel, and J. B. Schroder, *Least-squares finite element discretization of the neutron transport equation in spherical geometry*, SIAM J. Sci. Comput., 37 (2015), pp. S71–S89, https://doi.org/10.1137/140975152.
[31] O. E. Livne, *Coarsening by compatible relaxation*, Numer. Linear Algebra Appl., 11 (2004), pp. 205–227, https://doi.org/10.1002/nla.378.
[32] S. P. MacLachlan and L. N. Olson, *Theoretical bounds for algebraic multigrid performance: Review and analysis*, Numer. Linear Algebra Appl., 21 (2014), pp. 194–220, https://doi.org/10.1002/nla.1930.
[33] J. Mandel, M. Brezina, and P. Vaněk, *Energy optimization of algebraic multigrid bases*, Computing, 62 (1999), pp. 205–228, https://doi.org/10.1007/s006070050022.
[34] J. E. Morel and J. S. Warsa, *Spatial finite-element lumping techniques for the quadrilateral mesh S_n equations in x-y geometry*, Nuclear Sci. Eng., 156 (2007), pp. 325–342.
[35] A. Napov and Y. Notay, *Algebraic analysis of aggregation-based multigrid*, Numer. Linear Algebra Appl., 18 (2011), pp. 539–564, https://doi.org/10.1002/nla.741.
[36] Y. Notay, *An aggregation-based algebraic multigrid method*, Electron. Trans. Numer. Anal., 37 (2010), pp. 123–146.
[37] Y. Notay, *Algebraic analysis of two-grid methods: The nonsymmetric case*, Numer. Linear Algebra Appl., 17 (2010), pp. 73–96, https://doi.org/10.1002/nla.649.
[38] Y. Notay, *Algebraic theory of two-grid methods*, Numer. Math., 8 (2015), pp. 168–198, https://doi.org/10.4208/nmtma.2015.w04si.
[39] L. N. Olson, J. Schroder, and R. S. Tuminaro, *A new perspective on strength measures in algebraic multigrid*, Numer. Linear Algebra Appl., 17 (2010), pp. 713–733, https://doi.org/10.1002/nla.669.
[40] L. N. Olson and J. B. Schroder, *Components of a more robust multilevel solver for emerging architectures and complex applications*, in Proceedings of SciDAC 2011, U.S. Department of Energy, Washington, DC, 2011.
[41] 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, https://doi.org/10.1137/100803031.
[42] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, Frontiers in Appl. Math. 3, SIAM, Philadelphia, 1987, pp. 73–130.
[43] M. Sala and R. S. Tuminaro, *A new Petrov–Galerkin smoothed aggregation preconditioner for nonsymmetric linear systems*, SIAM J. Sci. Comput., 31 (2008), pp. 143–166, https://doi.org/10.1137/060659545.
[44] S. Schaffer, *A semicoarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients*, SIAM J. Sci. Comput., 20 (1998), pp. 228–242, https://doi.org/10.1137/S1064827595281587.
[45] J. B. Schroder, *Smoothed aggregation solvers for anisotropic diffusion*, Numer. Linear Algebra Appl., 19 (2012), pp. 296–312, https://doi.org/10.1002/nla.1805.
[46] P. Vaněk, M. Brezina, and J. Mandel, *Convergence of algebraic multigrid based on smoothed aggregation*, Numer. Math., 88 (2001), pp. 559–579, https://doi.org/10.1007/s211-001-8015-y.
[47] P. Vanek, A. Janka, and H. Guillard, *Convergence of Algebraic Multigrid Based on Smoothed Aggregation* II: *Extension to a Petrov-Galerkin Method*, Technical Report RR-3683, INRIA, Le Chesnay, France, 1999, https://hal.inria.fr/inria-00072986.
[48] P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems*, Computing, 56 (1996), pp. 179–196.
[49] P. S. Vassilevski, *Multilevel Block Factorization Preconditioners: Matrix-based Analysis and Algorithms for Solving Finite Element Equations*, Springer, New York, 2008, https://doi.org/10.1007/978-0-387-71564-3.
[50] T. A. Wiesner, R. S. Tuminaro, W. A. Wall, and M. W. Gee, *Multigrid transfers for nonsymmetric systems based on Schur complements and Galerkin projections*, Numer. Linear Algebra Appl., 21 (2014), pp. 415–438, https://doi.org/10.1002/nla.1889.