Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ORIGINAL RESEARCH

Generalizing Reduction-Based Algebraic Multigrid

Tareq Zaman*[1] | Nicolas Nytko[2] | Ali Taghibakhshi[3] | Scott MacLachlan[1] | Luke Olson[2] | Matthew West[3]

[1]Interdisciplinary Program in Scientific Computing, Memorial University of Newfoundland, NL, Canada
[2]Department of Computer Science, University of Illinois at Urbana-Champaign, Illinois, USA
[3]Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Illinois, USA

Correspondence
*T. U. Zaman, Interdisciplinary Program in Scientific Computing, Memorial University of Newfoundland, Newfoundland and Labrador, Canada. Email: tzaman@mun.ca

Abstract

Algebraic Multigrid (AMG) methods are often robust and effective solvers for solving the large and sparse linear systems that arise from discretized PDEs and other problems, relying on heuristic graph algorithms to achieve their performance. Reduction-based AMG (AMGr) algorithms attempt to formalize these heuristics by providing two-level convergence bounds that depend concretely on properties of the partitioning of the given matrix into its fine- and coarse-grid degrees of freedom. MacLachlan and Saad (SISC 2007) proved that the AMGr method yields provably robust two-level convergence for symmetric and positive-definite matrices that are diagonally dominant, with a convergence factor bounded as a function of a coarsening parameter. However, when applying AMGr algorithms to matrices that are not diagonally dominant, not only do the convergence factor bounds not hold, but measured performance is notably degraded. Here, we present modifications to the classical AMGr algorithm that improve its performance on matrices that are not diagonally dominant, making use of strength of connection, sparse approximate inverse (SPAI) techniques, and interpolation truncation and rescaling, to improve robustness while maintaining control of the algorithmic costs. We present numerical results demonstrating the robustness of this approach for both classical isotropic diffusion problems and for non-diagonally dominant systems coming from anisotropic diffusion.

KEYWORDS:
Algebraic Multigrid, Reduction-based Multigrid, Sparse Approximate Inverse.

1 | INTRODUCTION

Partial differential equations (PDEs) arise naturally as mathematical models of physical systems in many fields of science and engineering. As analytical techniques for their solution are limited to simple equations and geometries, numerical approximation of solutions via discretization techniques is ubiquitous. Standard discretizations necessitate the solution of large linear and nonlinear systems of equations which, in turn, requires fast and efficient algorithms. Among the techniques most commonly used for discretized elliptic equations are multigrid (MG) methods, known for their efficient and robust solution of a wide range of problems. Geometric multigrid (GMG) methods are often most efficient, but require detailed knowledge of the problem to be solved, its discretization, and regular structure of the underlying mesh hierarchy. In contrast, algebraic multigrid (AMG) methods can be effectively applied to problems on unstructured grids, or with highly variable (or discontinuous) coefficients. While the idea of AMG was first proposed over 40 years ago [1–4], understanding and improving the convergence of AMG remains an active area of research.

Classical (Ruge-Stüben) AMG has become a workhorse algorithm in scientific computing, particularly due to high-quality implementations available in standard packages[5–7]. While it provides an efficient and robust solution algorithm for a wide class of diffusion problems, its reliance on heuristics and algorithmic parameters (that can be difficult to tune) is often seen as a difficult hurdle to overcome, particularly for critical applications. As a result, significant effort has been invested in recent years in the development of AMG approaches with rigorous convergence bounds. Within this area are several approaches based on the pairwise aggregation methodology[8–12] that offers guaranteed convergence for problems such as the graph Laplacian. An alternative approach builds on the reduction-based multigrid methodology first proposed by Ries, Trottenberg, and Winter[13]. The reduction-based algebraic multigrid (AMGr) methodology introduced by MacLachlan et al.[14] uses algebraic properties of the linear system to determine reduction-like grid-transfer operators and relaxation that again leads to guaranteed convergence rates.

The basic principle of reduction-based multigrid follows from classical cyclic reduction algorithms[15] that reduce the cost of the direct solution of linear systems, $\boldsymbol{Ax} = \boldsymbol{b}$, by partitioning the degrees of freedom into two sets that we denote by F and C (in typical multigrid notation). The key feature of cyclic reduction is that this partitioning should be done in a way so that the submatrix of A over the set F, denoted A_{FF}, is a diagonal matrix. (Equivalently, the set F is an *independent set* in the graph associated with the sparse matrix, A.) Multigrid reduction[13] and AMGr[14] generalize this by allowing A_{FF} to be non-diagonal (F is not required to be an independent set), but only spectrally equivalent to a diagonal matrix. A key question left unanswered in this work is how to generate such partitions. MacLachlan and Saad[16] show that the task of generating the largest possible F set is an integer linear programming problem and, consequently, proposed a greedy algorithm for the partitioning with linear complexity. Notably, they proved that if A is symmetric, positive definite, and diagonally dominant, then there is a two-grid AMGr method with a guaranteed convergence bound. Several theoretical and practical improvements to the theory and algorithms of MacLachlan and Saad[16] have been proposed, with improved AMGr algorithms and convergence bounds[17, 18] and improved coarsening algorithms[19, 20]. Nonsymmetric variants have also been considered[21–23]. Further research into multigrid reduction (but not AMGr) has also been commonplace in recent years, with the emergence of multigrid-reduction-in-time[24] (MGRIT) methodologies for the solution of time-dependent PDEs.

Despite these developments, attaining effective AMGr convergence has remained essentially limited to systems with matrices that are either close to diagonally dominant or can be reordered to be close to lower triangular. Unfortunately, these limitations are significant, and preclude applying AMGr to many important and interesting classes of problems, including common AMG test cases, such as anisotropic diffusion equations, or problems discretized on anisotropic meshes. Poor performance on anisotropic problems, in particular, is reported in existing AMGr results[16, 19] that has only been overcome with expensive and impractical fixes, such as moving substantial numbers of points from F to C, in order to construct suitably improved interpolation operators within the AMGr framework. In this paper, we revisit the basic AMGr framework, with the goal of overcoming the barriers to achieving acceptable convergence for anisotropic problems. To do this, we look for more practical algorithmic choices within an AMGr-style algorithm. Specifically, we consider four ingredients for improving AMGr performance:

1. C-relaxation; while original AMGr focused on F-relaxation, we show that using C-relaxation (as considered in other settings[18, 24]) can be particularly helpful for these problems;

2. Sparse Approximate Inverses (SPAI); while the original AMGr papers focused on approximating A_{FF}^{-1} by a diagonal matrix, we consider using the SPAI algorithm[25–27] to offer better approximation of A_{FF}^{-1} while still maintaining sparsity;

3. Strength of Connection; while the original AMGr algorithms do not rely on the classical AMG[3, 4] notion of "strong connections" to filter small entries from the matrix, we find that such filtering is critical to success for anisotropic problems, where large "wrong-sign" off-diagonal entries appear, but cannot be productively used in the relaxation or coarse-grid correction processes; and

4. Interpolation truncation; the combination of techniques described above leads to effective AMGr-style solvers for a wider range of problems, but with higher algorithmic complexity than is needed; thus, we use the technique of interpolation truncation[28–30] to control these costs.

Numerical results demonstrate that these techniques can be used in combination to improve the performance of AMGr for both isotropic and anisotropic diffusion problems.

Sparse approximate inverses are one of a class of algorithms that define preconditioners for $\boldsymbol{Ax} = \boldsymbol{b}$ by prescribing a form for matrix M, then minimizing some norm of $I - MA$ (or $I - AM$). Originally proposed and investigated by Benson and Frederickson[31–33], recent investigations include factorized sparse approximate inverses[25] (FSAI), which aim to compute a sparse

Algorithm 1 AMG Setup Phase

```
function AMG-TWO-LEVEL-SETUP(A)
    C, F ← split the degrees of freedom into coarse and fine nodes
    P ← form interpolation operator
    A_c ← P^T A P
    return (P, A_c)
end function
```

approximation to the Cholesky factorization of SPD matrix A, and SPAI techniques[26, 27, 34] that directly compute sparse approximations to A^{-1}. The use of SPAI techniques in multigrid methods dates back almost to their initial introduction[31, 35], primarily to replace the use of standard relaxation schemes, such as the weighted Jacobi and Gauss-Seidel iterations. The use of SPAI techniques for relaxation within both geometric and algebraic multigrid has been considered more recently in several ways[36–42]. Similar ideas have been used in other contexts, to build interpolation operators[43, 44] or improve coarse-grid operators[45]. The work in this paper is closest to the ideas presented by Bollhöfer[46], where SPAI was used to determine both the relaxation scheme and the interpolation operator, although the remaining details of the scheme are quite different. We also note similarity to the work of Meurant[47, 48], where entries from the AINV[49] preconditioner were directly used to determine interpolation alongside AINV for relaxation.

The remainder of this paper is organized as follows. In Section 2, we give an introduction to algebraic multigrid, with a particular focus on reduction-based AMG (AMGr). We highlight, in Section 2.3, that AMGr as it exists has significant difficulties for anisotropic diffusion equations, motivating the work that follows. A key component of the algorithms considered here is the Sparse Approximate Inverse methodology of Grote and Huckle[27], we review this as well in Section 3. The main contribution of this paper is the generalized AMGr algorithm developed in Section 4. Supporting numerical results are presented in Section 5, followed by conclusions in Section 6.

2 | ALGEBRAIC MULTIGRID

Multigrid methods are based on the principle of complementarity, using fine-grid relaxation and coarse-grid correction to efficiently damp all errors in the approximation of solutions to linear systems $A\boldsymbol{x} = \boldsymbol{b}$. Geometric multigrid methods (GMG) fix a multigrid hierarchy by directly discretizing the PDE on a series of meshes defined by the problem geometry, and by adapting the relaxation scheme to complement the coarse-grid correction process defined in this way. Algebraic multigrid methods, in contrast, do not rely on explicit knowledge of the geometry nor the PDE, instead determining the coarse levels of the multigrid hierarchy in a setup phase that precedes the solution phase of the multigrid algorithm. In the setup, the set of degrees of freedom (or points) on the finest grid, Ω, is partitioned into disjoint sets, $\Omega = C \cup F$ (with $C \cap F = \emptyset$). The degrees of freedom in the set C constitute the points on the second level. Along with this partitioning, an interpolation operator, P, is constructed to map vectors from C onto Ω; similarly a restriction operator is defined, $R = P^T$ (in the symmetric case, as considered here), to map vectors from Ω onto C. With this, the Galerkin coarse-grid operator, $A_C = P^T A P$, is formed and the process continues recursively on A_C and C. The hierarchy is constructed until the number of nodes on a coarse grid is sufficiently small that direct factorization of A_C is feasible. The algorithm for a two-level setup phase is shown in Algorithm 1. Once the setup phase is completed, the solution phase solves the original system of equations using a standard multigrid cycling algorithm. A two-level algorithm is shown in Algorithm 2. Multilevel generalizations come from recursively solving $A_C \boldsymbol{e}_C = \boldsymbol{r}_C$ using the two-grid methodology, either once per level (leading to a V-cycle) or multiple two-grid sweeps per level (leading, for example, to the W-cycle). AMG methods are generally distinguished by how they define C from Ω, and how they define P from C and A. Below, we review the classical AMG algorithm, as presented by Ruge and Stüben[4], and the reduction-based AMG algorithm of MacLachlan et al.[14].

2.1 | Classical (Ruge-Stüben) AMG

Classical AMG makes use of the notion of *strong connections* in the graph corresponding to matrix A to define the coarse-fine partitioning. In the original algorithm, point i is said to be strongly connected to point j if $-A_{ij} \geq \gamma \max_{k \neq i}(-A_{ik})$ for some $\gamma \in (0, 1]$ (where the negative signs reflect the expectation that A be an M-matrix, or one where positive off-diagonal entries are

Algorithm 2 AMG Solution Phase

function AMG-TWO-LEVEL-V-CYCLE($A, \boldsymbol{b}, \boldsymbol{x}, P$)
  for $j \leftarrow 1, \ldots, \nu_1$ **do** ▷ Run ν_1 sweeps of pre-relaxation
    $\boldsymbol{x} \leftarrow$ relax on $\boldsymbol{x}$
  end for
  $\boldsymbol{r}_C \leftarrow P^T(\boldsymbol{b} - A\boldsymbol{x})$
  $\boldsymbol{e}_C \leftarrow$ solution of $A_C \boldsymbol{e}_C = \boldsymbol{r}_C$ ▷ Solve the coarse-level problem
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + P\boldsymbol{e}_C$
  for $j \leftarrow 1, \ldots, \nu_2$ **do** ▷ Run ν_2 sweeps of post-relaxation
    $\boldsymbol{x} \leftarrow$ relax on $\boldsymbol{x}$
  end for
  return $\boldsymbol{x}$
end function

not substantial). Many other definitions of strong connections have been considered in the literature, including the symmetric strength measure commonly used in smoothed aggregation AMG, where the connection between i and j is said to be strong if $|A_{ij}| \geq \gamma \sqrt{A_{ii} A_{jj}}$. Local approximations of the inverse of A and algebraically smooth error have also been used to define strong connections[50, 51], as has the concept of algebraic distances[52]. In all cases, the strength measure is used to "filter" the entries in A, and then the set C is defined by choosing a maximal independent set over the graph of strong connections. This can be characterized, for example, by the properties that each point in F is strongly dependent on at least one C-point, but no two C-points can be strongly dependent on one another.

The standard interpolation operator in classical AMG is formulated based on the assumption that algebraically smooth errors have small residuals after relaxation. Hence, the residual equation after relaxation can be written as $A\boldsymbol{e} \approx \boldsymbol{0}$. For the i-th degree of freedom, $i \in F$, this gives

$$A_{ii} e_i \approx -\sum_{k \in C_i} A_{ik} e_k - \sum_{j \in F_i^s} A_{ij} e_j - \sum_{\ell \in F_i^w} A_{i\ell} e_\ell, \tag{1}$$

where C_i, F_i^s, and F_i^w are the C-neighbors, strongly connected F-neighbors, and weakly connected F-neighbors of the i-th point, respectively. The standard interpolation operator is then derived by assuming that $e_\ell \approx e_i$ for weakly-connected neighbors, while

$$e_j \approx \frac{\sum_{k \in C_i} A_{jk} e_k}{\sum_{m \in C_i} A_{jm}},$$

for points $j \in F_i^s$. Substituting these into (1) and solving for e_i yields the interpolation formula

$$e_i = \sum_{k \in C_i} \frac{-A_{ik} - \sum_{j \in F_i^s} \frac{A_{ij} A_{jk}}{\sum_{m \in C_i} A_{jm}}}{A_{ii} + \sum_{\ell \in F_i^w} A_{i\ell}} e_k. \tag{2}$$

This defines the interpolation formula for all points $i \in F \subset \Omega$, assuming that e_k is known for all $k \in C_i$. For points $k \in C$, we use direct injection of values from coarse-grid points to their fine-grid counterparts in Ω.

2.2 | Reduction-based algebraic multigrid (AMGr)

Cyclic reduction[15] was originally proposed as a direct solver for certain linear systems that arose from finite-difference discretization of simple PDEs. Assuming that the degrees of freedom are already partitioned into coarse and fine nodes, the linear system $A\boldsymbol{x} = \boldsymbol{b}$ is reordered to have F degrees of freedom followed by C degrees of freedom, writing

$$A = \begin{bmatrix} A_{FF} & -A_{FC} \\ -A_{CF} & A_{CC} \end{bmatrix} \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_F \\ \boldsymbol{x}_C \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_F \\ \boldsymbol{b}_C \end{bmatrix}. \tag{3}$$

An *exact* algorithm for the solution of $A\boldsymbol{x} = \boldsymbol{b}$ in this partitioned form is given by

1. $\boldsymbol{y}_F = A_{FF}^{-1} \boldsymbol{b}_F$,
2. Solve $\left(A_{CC} - A_{CF} A_{FF}^{-1} A_{FC}\right) \boldsymbol{x}_C = \boldsymbol{b}_C + A_{CF} \boldsymbol{y}_F$,

3. $\boldsymbol{x}_F = \boldsymbol{y}_F + A_{FF}^{-1} A_{FC} \boldsymbol{x}_C$.

This can be turned into an iterative method for solving $A\boldsymbol{x} = \boldsymbol{b}$ in the usual way, replacing the right-hand side vector, $\boldsymbol{b}$, by the evolving residual, leading to reduction-based multigrid [13]. In this form, we compute updates to the current approximation, $\boldsymbol{x}^{(k)}$, as

1. $\boldsymbol{x}_F^{(k+1/2)} = \boldsymbol{x}_F^{(k)} + A_{FF}^{-1}\left(\boldsymbol{b}_F - A_{FF}\boldsymbol{x}_F^{(k)} + A_{FC}\boldsymbol{x}_C^{(k)}\right)$,
2. Solve $\left(A_{CC} - A_{CF}A_{FF}^{-1}A_{FC}\right)\boldsymbol{y}_C = \boldsymbol{b}_C + A_{CF}\boldsymbol{x}_F^{(k+1/2)} - A_{CC}\boldsymbol{x}_C^{(k)}$,
3. $\boldsymbol{x}_C^{(k+1)} = \boldsymbol{x}_C^{(k)} + \boldsymbol{y}_C$,
4. $\boldsymbol{x}_F^{(k+1)} = \boldsymbol{x}_F^{(k+1/2)} + A_{FF}^{-1}A_{FC}\boldsymbol{y}_C$.

We note that, as given above, this is still an exact algorithm, as $\boldsymbol{x}^{(1)} = \boldsymbol{x} = A^{-1}\boldsymbol{b}$ for any initial guess, $\boldsymbol{x}^{(0)}$. To make an iterative method from this skeleton, we introduce approximations of A_{FF}^{-1} in three places in the above algorithm, namely

$$\tilde{A}_{FF}^{-1} \approx A_{FF}^{-1} \qquad \tilde{A}_C \approx A_{CC} - A_{CF}A_{FF}^{-1}A_{FC} \qquad W_{FC} \approx A_{FF}^{-1}A_{FC}, \tag{4}$$

leading to the iteration:

1. $\boldsymbol{x}_F^{(k+1/2)} = \boldsymbol{x}_F^{(k)} + \tilde{A}_{FF}^{-1}\left(\boldsymbol{b}_F - A_{FF}\boldsymbol{x}_F^{(k)} + A_{FC}\boldsymbol{x}_C^{(k)}\right)$,
2. Solve $\tilde{A}_C\boldsymbol{y}_C = \boldsymbol{b}_C + A_{CF}\boldsymbol{x}_F^{(k+1/2)} - A_{CC}\boldsymbol{x}_C^{(k)}$,
3. $\boldsymbol{x}_C^{(k+1)} = \boldsymbol{x}_C^{(k)} + \boldsymbol{y}_C$,
4. $\boldsymbol{x}_F^{(k+1)} = \boldsymbol{x}_F^{(k+1/2)} + W_{FC}\boldsymbol{y}_C$.

Viewing this as a two-grid algorithm, we recognize the first step as special form of relaxation, known as F-relaxation, where the approximation to A_{FF}^{-1} is accomplished via a standard weighted Jacobi or Gauss-Seidel iteration. The second step then represents a coarse-grid solve, where the residual is restricted to the coarse-grid by injection, and the correction, $\boldsymbol{y}_C$, is computed using an approximation, $\tilde{A}_C$, of the true Schur complement, $A_{CC} - A_{CF}A_{FF}^{-1}A_{FC}$. The final two steps represent the interpolation of the correction, writing the interpolation operator $P = \begin{bmatrix} W_{FC} \\ I \end{bmatrix}$. This can be viewed as an approximation of the *ideal* interpolation operator [53], $W_{FC} \approx A_{FF}^{-1}A_{FC}$. Notably, this algorithm differs from standard multigrid cycling in several ways, including the fixed use of injection for the restriction of the residual to the coarse grid, and the lack of post-relaxation sweeps.

As written above, there is little guidance in how to choose the three approximations in (4). MacLachlan et al. [14] address this in their development of the reduction-based AMG (AMGr) algorithm, connecting convergence of the two-grid scheme with properties of A_{FF}. In particular, it is assumed that A_{FF} can be approximated by known matrix D_{FF} for which computing the action of D_{FF}^{-1} on a vector is computationally feasible. From this, Theorem 1 holds, using the notation that matrices $A \succeq B$ when $\boldsymbol{x}^T A\boldsymbol{x} \geq \boldsymbol{x}^T B\boldsymbol{x}$ for all vectors $\boldsymbol{x}$.

Theorem 1. [14] Consider the symmetric and positive-definite matrix $A = \begin{bmatrix} A_{FF} & -A_{FC} \\ -A_{FC}^T & A_{CC} \end{bmatrix}$ such that $A_{FF} = D_{FF} + \mathcal{E}$, with D_{FF} symmetric, $0 \preceq \mathcal{E} \preceq \epsilon D_{FF}$ for some $\epsilon \geq 0$, and $\begin{bmatrix} D_{FF} & -A_{FC} \\ -A_{FC}^T & A_{CC} \end{bmatrix} \geq 0$. Define relaxation with error-propagation operator $R = \left(I - \sigma \begin{bmatrix} D_{FF}^{-1} & 0 \\ 0 & 0 \end{bmatrix} A\right)$ for $\sigma = 2/(2+\epsilon)$, interpolation $P = \begin{bmatrix} D_{FF}^{-1}A_{FC} \\ I \end{bmatrix}$, and coarse-level correction with error-propagation operator $T = I - P(P^TAP)^{-1}P^TA$. Then the multigrid cycle with ν pre-relaxation sweeps, coarse-level correction, and ν post-relaxation sweeps has error propagation operator $\mathrm{MG}_2 = R^\nu \cdot T \cdot R^\nu$ which satisfies

$$||\mathrm{MG}_2||_A \leq \left(\frac{\epsilon}{1+\epsilon}\left(1 + \left(\frac{\epsilon^{2\nu-1}}{(2+\epsilon)^{2\nu}}\right)\right)\right)^{1/2} < 1. \tag{5}$$

Several generalizations of both the theory and practice of reduction-based multigrid methods have since been developed. A generalization to non-symmetric M-matrices was proposed and analyzed by Mense and Nabben [54], using the tools of weak regular splitting [55]. For symmetric and positive definite problems, Brannick et al. [18] study the introduction of more general relaxation schemes, as well as the use of different approximations of A_{FF} for interpolation and relaxation. Gossler and Nabben [17] examine generalization of AMGr to the use of Chebyshev polynomial acceleration of multiple relaxation sweeps. For strongly non-symmetric systems, Manteuffel et al. [22, 23] have proposed similar approaches using so-called approximate ideal restriction (AIR) techniques, that offer excellent performance for advection-dominated problems. None of the above schemes, however, address the poor performance observed in AMGr-type methods for anisotropic problems, which is the motivation for the present work.

2.2.1 | Coarsening in AMGr

Theorem 1 establishes existence of an interpolation operator, P, provided that the matrix A can be partitioned into $\begin{bmatrix} A_{FF} & -A_{FC} \\ -A_{FC}^T & A_{CC} \end{bmatrix}$ and an approximation, D_{FF}, of A_{FF} can be made that satisfies the assumptions in the theorem. While this is insightful, it does not address the fundamental question of how to generate a partitioning for which the assumptions hold with small parameter ϵ. To answer this question, MacLachlan and Saad[16] propose to partition the rows and columns of A in order to ensure the diagonal dominance of A_{FF}, allowing D_{FF} to be chosen as a diagonal matrix. In particular, for each row, i, the diagonal dominance of row i over the F points is quantified by

$$\eta_i = \frac{|A_{ii}|}{\sum_{j \in F} |A_{ij}|}.$$

Then, A_{FF} is said to be η-diagonally dominant if $\eta_i \geq \eta$ for all $i \in F$, for some $\eta > 1/2$ that measures the diagonal dominance of A_{FF}. If A_{FF} is η-diagonally dominant, then the diagonal matrix, D_{FF}, with $(D_{FF})_{ii} = (2 - \frac{1}{\eta})A_{ii}$ for all $i \in F$ yields $0 \preceq \mathcal{E} \preceq \frac{2-2\eta}{2\eta-1} D_{FF}$, giving an η-dependent convergence bound if the other assumptions of Theorem 1 are satisfied. Furthermore, if A is symmetric, positive-definite, and diagonally dominant, then this condition guarantees that all conditions of Theorem 1 are satisfied.

In addition to establishing this connection between the diagonal dominance parameter η and the convergence parameter, ϵ, MacLachlan and Saad[16] pose the partitioning algorithm as an optimization problem: for a given $\eta > 1/2$, find the largest F-set such that $\eta_i \geq \eta$ for every $i \in F$. This can be written as

$$\begin{aligned} &\max_{F \subset \Omega} |F|, \\ &\text{subject to } |A_{ii}| \geq \eta \sum_{j \in F} |A_{ij}|, \ \forall i \in F. \end{aligned} \tag{6}$$

MacLachlan and Saad[16] show that finding the optimal solution to (6) is NP-complete and, consequently, propose a greedy algorithm to approximately solve the optimization problem. The greedy algorithm acts iteratively, adding points to the C-set one at a time, and moving any points that are guaranteed to satisfy the inequality constraint in (6) into the F-set, until a full partition is computed.

While the greedy coarsening algorithm was demonstrated to be effective in some settings, Zaman et al.[19] demonstrate that there are also cases where the resulting optimality gap can be significant. To address this, they propose to apply simulated annealing to the optimization problem in (6). In this approach, the set of points, Ω, is first partitioned into non-overlapping subdomains, and a local partitioning is computed on each subdomain, starting from the assumption that all points are to be made C points. Using a Gauss-Seidel-like approach, the subdomains are processed successively, running a number of annealing steps on points within the subdomain, randomly swapping some C points to be F points or vice-versa, and accepting the move (with some probability) if it improves the overall fitness (quantified by the number of F points that satisfy the constraint in (6)). Zaman et al.[19] show that this approach can produce substantially better partitionings than the greedy approach, albeit at the greatly increased cost of many simulated annealing steps. Similar work by Taghibakhshi et al.[20] uses reinforcement learning to solve the same problem at a lower cost. In both of these papers, while "good" solutions are found to the optimization problem in (6), poor performance is observed for anisotropic diffusion problems, as these problems do not satisfy the diagonal dominance condition required by the theory of MacLachlan and Saad[16] in the assumptions of Theorem 1.

2.3 | Failure of AMGr for anisotropic diffusion

To demonstrate the convergence problems, we consider applying AMGr (following the prescription of MacLachlan and Saad[16]) to the solution of the two-dimensional anisotropic diffusion problem,

$$-\nabla \cdot \boldsymbol{K}(x, y) \nabla u(x, y) = b(x, y) \tag{7}$$

in the domain $[0, 1] \times [0, 1]$ with Dirichlet boundary conditions. We choose the tensor coefficient $\boldsymbol{K}(x, y) = QHQ^T$, where

$$Q = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \qquad H = \begin{bmatrix} 10^{-6} & 0 \\ 0 & 1 \end{bmatrix}, \tag{8}$$

and θ specifies the direction of anisotropy in the problem. For $\theta = 0$ this gives the grid-aligned anisotropic equation $-10^{-6} u_{xx} - u_{yy} = b$, while $0 < \theta < \pi/2$ gives a non-grid-aligned diffusion tensor. Table 1 presents convergence results for the standard finite-difference and bilinear finite element discretizations of this problem, with $\theta = 0$, $\pi/6$, and $\pi/4$. For simplicity, we present

TABLE 1 Performance of two-level AMGr for the anisotropic diffusion problem.

θ	Discretization	Eigenvalues of $D_{FF}^{-1}A_{FF}$ min	Eigenvalues of $D_{FF}^{-1}A_{FF}$ max	$\begin{bmatrix} D_{FF} & -A_{FC} \\ -A_{FC}^T & A_{CC} \end{bmatrix}$	Convergence factor ρ	Complexities C_{grid}	Complexities C_{op}
0	FD	1.00	3.00	positive definite	0.75	1.33	1.49
	FE	1.00	6.83	indefinite	0.98	1.49	2.14
$\pi/6$	FD	1.04	5.59	indefinite	0.96	1.42	1.87
	FE	1.00	7.09	indefinite	0.97	1.36	1.72
$\pi/4$	FD	1.21	6.73	indefinite	0.96	1.39	1.79
	FE	1.12	5.80	indefinite	0.96	1.33	1.59

results for a uniform 32×32 grid, although similar results are observed for larger meshes. Here, and in all results that follow, we measure convergence by solving the homogeneous problem, $A\boldsymbol{x} = \boldsymbol{0}$, with a randomly chosen initial guess for $\boldsymbol{x}$. Writing $\boldsymbol{e}^{(k)}$ as the error in the k^{th} approximation to $\boldsymbol{x} = \boldsymbol{0}$, we estimate the asymptotic convergence factor by running 50 (stationary) multigrid iterations, then estimating $\rho \approx \left(\|\boldsymbol{e}^{(50)}\|_A / \|\boldsymbol{e}^{(10)}\|_A\right)^{1/40}$, averaging convergence over the final 40 iterations. We note that, in all cases, the coarsening algorithm (simulated annealing, in this case) generates a partitioning such that the matrix A_{FF} is well-approximated by a diagonal matrix, D_{FF}; however, only in the case of the finite-difference discretization of the grid-aligned diffusion equation (when the discretization matrix, A, *is* diagonally dominant), does the required semidefiniteness of $\begin{bmatrix} D_{FF} & -A_{FC} \\ -A_{FC}^T & A_{CC} \end{bmatrix}$ hold. This correlates strongly with the resulting measured asymptotic convergence factor for the method. Both MacLachlan and Saad[16] and Zaman et al.[19] consider remedies for this behavior, such as augmenting the C set in a style similar to classical AMG; while this improves the overall convergence of the method, it also leads to greatly increased grid and operator complexities, making it an unsatisfactory solution.

Here, and in all tables that follow, we use color-coding to indicate quality of the results shown. For measured convergence factors, we denote a "good" convergence factor to be below 0.4 (indicated in green), while a "bad" convergence factor, above 0.8, is shown in red (with values in between, $0.4 < \rho < 0.8$, shown in black text). For two-grid operator complexity, we highlight results in green if the complexity is below 1.5, in red if it is above 2.5, and in orange for values between 2.0 and 2.5. As operator complexity, in particular, is expected to grow with the number of levels in the hierarchy, we use similar highlighting with different thresholds for three-grid and multigrid operator complexity, showing results in green if it is below 2.0, red if it is above 3.0, and orange for values between 2.5 and 3.0. As the results that follow show relatively little variation in AMG grid complexity, we choose not to highlight values for this measure. Definitions of these complexities are found below, in Section 4.1.

3 | SPARSE APPROXIMATE INVERSE (SPAI) METHODS

While originally proposed in the 1970's by Benson and Frederickson[31, 32], SPAI techniques were more systematically developed and studied in the 1990's (and subsequently) by a number of authors[25–27, 34, 56]. The general idea of SPAI techniques is to compute a matrix, M, to minimize some norm of $I - MA$ or $I - AM$, with constraints on the sparsity of M. These constraints may be fixed (e.g., some fixed set of elements of M is allowed to be nonzero), or may be adaptively determined by trying to best minimize the chosen norm within some limitations on either the total number of nonzero entries in M or the row/column-wise number of nonzero elements. Here, we focus on the variant of the SPAI algorithm proposed by Hawkins and Chen[57], in which the Frobenius norm of $B - AM$ is minimized for a given matrix, B, over a fixed nonzero pattern for each column. If $B = I$, then this reduces to a simplified version of the SPAI algorithm of Grote and Huckle[27], omitting their adaptive calculation for increasing the nonzero pattern for each column.

Algorithm 3 presents the SPAI algorithm, where the inputs are given by matrices A and B, and a nonzero sparsity pattern, $\mathcal{S}$, for the sparse approximate inverse M. The algorithm loops independently over each column, j, in M. In the initialization stage of the algorithm (Lines 3 through 7), the rows, $\mathcal{J}$, in the initial sparsity pattern of $\mathcal{S}$ for column j are extracted, as is the set of rows, $\mathcal{I}$, of A for which matrix A has a nonzero entry in a column in $\mathcal{J}$, defining a submatrix, $\bar{A}$, of A that is used to initialize column j of M. Two auxiliary vectors are also formed, corresponding to the full j^{th} column of B, denoted $\boldsymbol{b}$, and its restriction to the rows of $\bar{A}$, denoted $\bar{\boldsymbol{b}}$. Column j of M then comes from using the QR decomposition of $\bar{A}$ to solve the unconstrained

minimization problem of minimizing $\|\bar{\boldsymbol{b}} - \bar{A}\bar{\boldsymbol{m}}\|_2$, noting that $\bar{A}$ is expected, by its construction, to have more rows than columns, so that this is not expected to yield a zero residual. The computed solution, $\bar{\boldsymbol{m}}$, is injected into a full vector, $\boldsymbol{m}$, which becomes the j^{th} column of M.

Algorithm 3 SPAI algorithm

function SPAI($A, B, \mathcal{S}$)
  for $j \leftarrow 1, \ldots, n_{\text{col}}$ **do** ▷ n_{col} is number of columns in A
    $\mathcal{J} \leftarrow \{i \mid (i,j) \in \mathcal{S}\}$
    $\mathcal{I} \leftarrow$ set of indices of nonzero rows of $A(:, \mathcal{J})$
    $\boldsymbol{b} \leftarrow B(:, j)$
    $\bar{A} \leftarrow A(\mathcal{I}, \mathcal{J})$
    $\bar{\boldsymbol{b}} \leftarrow B(\mathcal{I}, j)$
    Compute QR decomposition of $\bar{A}$
    $\bar{\boldsymbol{m}} \leftarrow \text{argmin}_{\bar{\boldsymbol{m}}} \|\bar{\boldsymbol{b}} - \bar{A}\bar{\boldsymbol{m}}\|_2$ ▷ Unconstrained least-squares via QR
    $\boldsymbol{m} \leftarrow \bar{\boldsymbol{m}}$ with inserted zeros
    $M(:, j) \leftarrow \boldsymbol{m}$
  end for
  return M
end function

Sparse approximate inverse algorithms similar to Algorithm 3 have been investigated for use in both relaxation and interpolation in several settings in the past. However, this usage has generally been in defining approximations to the inverse of A in its entirety, while we look at the possible use of SPAI techniques through the lens of the AMGr methodology. In what follows, we will make use of SPAI in three ways:

1. In F-relaxation where, given a proxy matrix, $\hat{A}_{FF}$, for A_{FF} (possibly equal to A_{FF}), and $B = I_{FF}$, D_{FF}^{inv} is constructed as the SPAI approximation to $\hat{A}_{FF}^{-1}$ with a fixed sparsity pattern equal to that of $\hat{A}_{FF}$;
2. In C-relaxation where, given a proxy matrix, $\hat{A}_{CC}$, for A_{CC} (possibly equal to A_{CC}), and $B = I_{CC}$, D_{CC}^{inv} is constructed as the SPAI approximation to $\hat{A}_{CC}^{-1}$ with a fixed sparsity pattern equal to that of $\hat{A}_{CC}$; and
3. In interpolation, where we use the Hawkins and Chen modfication[57], to solve $\min_W \|\hat{A}_{FC} - \hat{A}_{FF}W\|_F$ for sparse approximate columns of $W = \hat{A}_{FF}^{-1}\hat{A}_{FC}$ for proxy matrices, $\hat{A}_{FF}$ and $\hat{A}_{FC}$, for A_{FF} and A_{FC}, respectively, with a fixed nonzero pattern equal to that of $\hat{A}_{FC} + \hat{A}_{FF}\hat{A}_{FC}$.

The first two of these can be viewed as generalizations of the use of SPAI on all of A as relaxation[37,39] to the F- and C-relaxations typically used in reduction-based AMG. The third bears similarity to Meurant's Algorithm I3[47], where SPAI on A (either in its original ordering or reordered according to the F-C partitioning) is used to generate an approximate inverse matrix, M, from which M_{FF} is extracted to form an interpolation operator $\begin{bmatrix} M_{FF}A_{FC} \\ I \end{bmatrix}$. We note that this is akin to approximating ideal interpolation, $\begin{bmatrix} A_{FF}^{-1}A_{FC} \\ I \end{bmatrix}$ by using an approximation to $\left(A^{-1}\right)_{FF}$ rather than $\left(A_{FF}\right)^{-1}$. As discussed below, direct use of SPAI to approximate A_{FF}^{-1}, A_{CC}^{-1}, and $A_{FF}^{-1}A_{FC}$ in these contexts does not directly lead to effective performance, so we consider additional tools from standard AMG development to both improve convergence and lower cost.

4 | GENERALIZING AMGR

Baseline results: From the results in Table 1 and those documented in other works[16,19,20], the coarse-grid correction process emerges as a primary source for the poor performance of AMGr on anisotropic problems. Using $D_{FF}^{-1}A_{FC}$ with diagonal D_{FF} for the C-to-F interpolation matrix is more restrictive than the interpolation operators used in classical multigrid, as interpolation to an F point is only allowed from directly connected C points (corresponding to the nonzero entries in A_{FC}). To test this theory, we first use SPAI to determine an interpolation operator of the form $P = \begin{bmatrix} W \\ I \end{bmatrix}$, where the sparsity pattern of W is fixed

TABLE 2 Two-level AMGr convergence factors for anisotropic FE discretization using semi-coarsening in the y direction by a factor of three. Interpolation of F-nodes uses the SPAI approximation to $A_{FF}^{-1}A_{FC}$ and the SPAI approximation to A_{FF}^{-1} is used for relaxation.

	$\theta = 0$			$\theta = \pi/6$			$\theta = \pi/4$		
Grid size	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}
16×16	0.021	1.31	1.90	0.006	1.31	1.90	0.213	1.31	1.90
32×32	0.023	1.31	2.02	0.019	1.31	2.02	0.518	1.31	2.02
64×64	0.023	1.33	2.14	0.065	1.33	2.14	0.808	1.33	2.14
128×128	0.024	1.33	2.17	0.210	1.33	2.17	0.921	1.33	2.17

to match that of $A_{FC} + A_{FF}A_{FC}$, allowing for interpolation to a fine-grid point from both directly connected C points and C points that are directly connected to an adjacent F point. At the same time, we replace relaxation based on a diagonal stencil with relaxation using the SPAI approximation to A_{FF}^{-1} with the sparsity pattern of A_{FF}.

To test the effects of these modifications, we again consider the anisotropic diffusion equation given in (7), for three angles, $\theta = 0$, $\pi/6$, and $\pi/4$, with convergence factors shown in Table 2. To decouple the impact of these choices from that of the coarsening, we use a geometric coarse grid chosen as semi-coarsening by a factor of three in the y-direction (the direction of strong connections in these cases). We observe significant improvement in convergence in both the $\theta = 0$ and $\theta = \pi/6$ cases, in comparison to the results in Table 1, although performance for $\theta = \pi/4$ is much worse. Yet, we also note that the complexity of these cycles is high, with two-grid operator complexities above 2.0 due to many small nonzero entries in the resulting interpolation operators that lead to large numbers of nonzero entries in the Galerkin coarse-grid operator, $P^T AP$.

Strong connections: To address the high cost encountered in the baseline, we introduce strong connections into the algorithm. A typical row in the matrix for an anisotropic diffusion operator contains both small entries and large but "wrong-sign" entries, where there are positive contributions in directions other than the strong direction of diffusion in the PDE. In response, we introduce a filtering stage, where we compute a *proxy* matrix, $\hat{A}$, for the given system matrix, A. We first compute strong connections using the classical Ruge-Stüben definition of strength of connection, defining point i to be strongly connected to point j if

$$-A_{ij} \geq \frac{1}{2} \max_{k \neq i} -A_{ik}, \tag{9}$$

where a strength parameter of 1/2 is selected as is typical for anisotropic PDEs and where only negative off-diagonal entries are allowed as strong connections (also common practice).

For anisotropic diffusion equations discretized by bilinear finite elements on uniform grids, (9) results in two strong connections for each interior node, aligned vertically (north and south), for $\theta = 0$, two strong connections in the north-east and south-west directions for $\theta = \pi/4$, but four strong connections for $\theta = \pi/6$, including north, south, north-east, and south-west points. To preserve the row-sum that is typically needed for best AMG performance, we define $\hat{A}$ to have off-diagonal entries matching those of A for strong connections, and diagonal entries adjusted by subtracting any weak connections in each row of A from its diagonal value (so-called "lumping" of the weak connections to the diagonal, as in (2)).

Next, we repeat the experiments above, but define interpolation as the SPAI approximation to $\hat{A}_{FF}^{-1}\hat{A}_{FC}$ and use F-relaxation based on the SPAI approximation to $\hat{A}_{FF}^{-1}$, with results presented in Table 3. For $\theta = 0$ and $\theta = \pi/4$, each F-point has a single strongly-connected C neighbor and a single strongly-connected F neighbor, while each F-point for $\theta = \pi/6$ has two of each. Using semi-coarsening in the y-direction by a factor of three, this results in interpolation to each F-point from two C-points for $\theta = 0$ and $\pi/4$ and from five C-points for $\theta = \pi/6$ (where the two strongly connected F neighbors of an F-point have a total of three strongly connected C neighbors). From the table, we see that using this definition of strength results in improved and grid-independent convergence for the case of $\theta = \pi/4$, and degraded (but still grid-independent) convergence for $\theta = 0$. However, significant degradation in convergence occurs for the case of $\theta = \pi/6$. Nonetheless, the use of strong conenctions has greatly improved the two-grid operator complexities, particularly for the $\theta = 0$ case, where it now matches that of geometric multigrid.

Interpolation Scaling: Tables 2 and 3 underscore the potential of SPAI-based AMGr for anisotropic diffusion equations, but also highlight that acceptable and scalable convergence is not robust. From the poor convergence for $\theta = \pi/6$ in Table 3, we found that even if the "lumped" matrix, $\hat{A}$, used to form interpolation retains the property that rows away from boundary conditions have zero row sum (and that $\hat{A}$ is an M-matrix), the interpolation operator determined by SPAI does not accurately

TABLE 3 Two-level AMGr convergence factors for anisotropic FE discretization using semi-coarsening in the y direction by a factor of three. Interpolation of F-nodes is based on a SPAI approximation to $\hat{A}_{FF}^{-1}\hat{A}_{FC}$ and relaxation uses a SPAI approximation to $\hat{A}_{FF}^{-1}$, **where $\hat{A}$ is the "lumped" matrix of strong connections computed from A.**

	$\theta = 0$			$\theta = \pi/6$			$\theta = \pi/4$		
Grid size	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}
16×16	0.351	1.31	1.28	0.289	1.31	1.60	0.696	1.31	1.42
32×32	0.359	1.31	1.30	0.487	1.31	1.66	0.718	1.31	1.47
64×64	0.359	1.33	1.32	0.745	1.33	1.73	0.716	1.33	1.52
128×128	0.359	1.33	1.32	0.906	1.33	1.75	0.716	1.33	1.53

TABLE 4 Two-level AMGr convergence factors for anisotropic FE discretization using semi-coarsening in the y direction by a factor of 3. Here, F-nodes are interpolated using the SPAI approximation to $\hat{A}_{FF}^{-1}\hat{A}_{FC}$, **postprocessed to exactly interpolate the constant vector**, and the SPAI approximation to $\hat{A}_{FF}^{-1}$ is used for relaxation, where $\hat{A}$ is the "lumped" matrix of strong connections computed from A.

	$\theta = 0$			$\theta = \pi/6$			$\theta = \pi/4$		
Grid size	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}
16×16	0.751	1.31	1.28	0.641	1.31	1.60	0.765	1.31	1.42
32×32	0.776	1.31	1.30	0.640	1.31	1.66	0.751	1.31	1.47
64×64	0.772	1.33	1.32	0.649	1.33	1.73	0.744	1.33	1.52
128×128	0.772	1.33	1.32	0.656	1.33	1.75	0.741	1.33	1.53

interpolate the coarse-grid constant function onto the fine-grid. This is not surprising, since SPAI computes the interpolation operator column-wise, yet interpolation to any fixed fine-grid vector is a row-wise property of matrix P.

To address the lack of constant interpolation, we post-process the interpolation generated by SPAI, using left diagonal scaling of $W \approx \hat{A}_{FF}^{-1}\hat{A}_{FC}$ so that $\mathbf{1}_F = SW\mathbf{1}_C$. This is accomplished by computing $\boldsymbol{s} = W\mathbf{1}_C$, followed by defining diagonal matrix S with entries $1/s_i$ on its diagonal. This yields the convergence factors in Table 4. The results offer concrete improvement over Tables 2 and 3, in that they offer scalable convergence for all three problems (without increasing grid or operator complexities). Even so, the overall convergence factors between 0.65 and 0.8 are insufficient to be considered an effective AMG solver.

One possible cause of the degraded convergence is poor interpolation near Dirichlet boundaries, where the constant vector is not an accurate indicator of the slowest-to-converge modes of relaxation. As a remedy, we use a similar scaling of interpolation, that we refer to as "improved iteration" scaling, running a set number of sweeps of (full grid) weighted Jacobi relaxation on the homogeneous problem with the constant vector as an initial guess, to produce a relaxed vector, $\boldsymbol{z}$, and followed by a similar diagonal scaling computed to ensure that $\boldsymbol{z}_F = SW\boldsymbol{z}_C$. With this modification and five sweeps of relaxation, Table 5 shows notable improvement in performance for both the case of $\theta = 0$ and $\theta = \pi/6$, but still disappointing (albeit grid-independent) convergence for $\theta = \pi/4$.

C-relaxation: As a final modification we employ the use of C-relaxation alongside F-relaxation. As has been considered in MGRIT[24] and other contexts, replacing simple F-relaxation with sweeps of FCF-relaxation (that is, relaxation over the F-points, followed by relaxation over the C-points, then again over the F-points, with updated residual values between each sweep) is known to greatly improve multigrid performance in some settings. Results using FCF-relaxation are shown in Table 6, where C-relaxation is again computed with SPAI on $\hat{A}_{CC}$. We see that including C-relaxation results in a dramatic effect for $\theta = \pi/4$, reducing the convergence factor to nearly 0.1, and a notable effect for $\theta = 0$. For $\theta = \pi/6$, adding C-relaxation has little influence on convergence, noting it does not harm performance.

TABLE 5 Two-level AMGr convergence factors for anisotropic FE discretization using semi-coarsening in the y direction by a factor of 3. Here, F-nodes are interpolated using the SPAI approximation to $\hat{A}_{FF}^{-1}\hat{A}_{FC}$, **postprocessed to exactly interpolate a relaxed vector**, and the SPAI approximation to $\hat{A}_{FF}^{-1}$ is used for relaxation, where $\hat{A}$ is the "lumped" matrix of strong connections computed from A.

	$\theta = 0$			$\theta = \pi/6$			$\theta = \pi/4$		
Grid size	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}
16×16	0.379	1.31	1.28	0.177	1.31	1.60	0.720	1.31	1.42
32×32	0.382	1.31	1.30	0.197	1.31	1.66	0.719	1.31	1.47
64×64	0.379	1.33	1.32	0.256	1.33	1.73	0.717	1.33	1.52
128×128	0.377	1.33	1.32	0.284	1.33	1.75	0.717	1.33	1.53

TABLE 6 Two-level AMGr convergence factors for anisotropic FE discretization using semi-coarsening in the y direction by a factor of 3. Here, F-nodes are interpolated using the SPAI approximation to $\hat{A}_{FF}^{-1}\hat{A}_{FC}$, postprocessed to exactly interpolate a relaxed vector, and the SPAI approximations to $\hat{A}_{FF}^{-1}$ and $\hat{A}_{CC}^{-1}$ are used **for FCF-relaxation**, where $\hat{A}$ is the "lumped" matrix of strong connections computed from A.

	$\theta = 0$			$\theta = \pi/6$			$\theta = \pi/4$		
Grid size	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}
16×16	0.233	1.31	1.28	0.107	1.31	1.60	0.110	1.31	1.42
32×32	0.238	1.31	1.30	0.186	1.31	1.66	0.111	1.31	1.47
64×64	0.240	1.33	1.32	0.249	1.33	1.73	0.115	1.33	1.52
128×128	0.239	1.33	1.32	0.279	1.33	1.75	0.114	1.33	1.53

4.1 | Algebraic Coarsening

Given the satisfactory results in Table 6, we next focus on extending these results to fully algebraic coarsening using the simulated annealing coarsening described in Section 2.2.1. We note that this coarsening is computationally quite expensive [19], but that it provides the best known complexities for AMGr-style methods. In Section 5.4, we experiment with the more feasible greedy coarsening algorithm of MacLachlan and Saad [16]. An important consideration in assessing the quality of algebraic coarsening is the resulting complexity of the multigrid algorithm, as this is no longer determined *a priori* from the fixed geometric coarsening. We use two common measures: grid and operator complexity. The AMG grid complexity, C_{grid}, is the ratio of the sum of the number of DoFs on each level of multigrid hierarchy (including the finest) to that on the finest level. Similarly, the operator complexity, C_{op}, is the ratio of the sum of the number of nonzeros in the system matrices on each level of hierarchy (including the finest) to that on the finest level.

Algebraic coarsening: Tables 7 and 8 show the convergence factors and corresponding grid and operator complexities for two-grid cycles using two values of the diagonal dominance parameter in (6). From preliminary experiments (not reported here), we noted substantial improvement when computing the fine-coarse partitioning using $\hat{A}$ (compared with A); hence, we use $\hat{A}$ in all subsequent results.

Table 7 uses $\eta = 0.65$, resulting in two-level grid complexities, $C_{\text{grid}} \approx 4/3$, matching that of the geometric semi-coarsening by three used above. Using a larger parameter, $\eta = 0.75$, in Table 8, results in $C_{\text{grid}} \approx 3/2$, consistent with geometric semi-coarsening by a factor of two. The coarsening is visualized in Figure 1 for both cases and $\theta = 0$, demonstrating that, while the coarsening is still algebraic, it retains much of the geometric character of semi-coarsening. As expected, using a larger value of η leads to an improvement in convergence factors (since the coarse-grid correction is over a larger space), but also higher complexities. In particular, for the case of $\theta = 0$, we maintain complexities similar to those of geometric multigrid for these problems, with $C_{\text{op}} \approx C_{\text{grid}}$, but we also see the typical increase in AMG operator complexity faster than grid complexity for $\theta = \pi/6$ and $\pi/4$, indicating increased density in the coarse-grid operators. While the effective convergence factors, defined as $\rho^{1/C_{\text{op}}}$, are lower for $\eta = 0.75$ than $\eta = 0.65$, we emphasize that these are only two-grid complexities, and denser coarse-grid

TABLE 7 Two-level AMGr convergence factors and corresponding complexities for anisotropic FE discretization **using simulated annealing coarsening with** $\eta = 0.65$. F-nodes are interpolated using the SPAI approximation to $\hat{A}_{FF}^{-1}\hat{A}_{FC}$, postprocessed to exactly interpolate a relaxed vector, and the SPAI approximations to $\hat{A}_{FF}^{-1}$ and $\hat{A}_{CC}^{-1}$ are used for FCF-relaxation, where $\hat{A}$ is the "lumped" matrix of strong connections computed from A.

	$\theta = 0$			$\theta = \pi/6$			$\theta = \pi/4$		
Grid size	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}
16×16	0.219	1.32	1.32	0.110	1.35	1.69	0.116	1.30	1.40
32×32	0.235	1.32	1.33	0.189	1.35	1.78	0.114	1.32	1.49
64×64	0.234	1.34	1.37	0.342	1.36	1.91	0.121	1.33	1.54
128×128	0.231	1.34	1.39	0.400	1.36	1.96	0.133	1.34	1.57

TABLE 8 Two-level AMGr convergence factors and corresponding complexities for anisotropic FE discretization **using simulated annealing coarsening with** $\eta = 0.75$. F-nodes are interpolated using the SPAI approximation to $\hat{A}_{FF}^{-1}\hat{A}_{FC}$, postprocessed to exactly interpolate a relaxed vector, and the SPAI approximations to $\hat{A}_{FF}^{-1}$ and $\hat{A}_{CC}^{-1}$ are used for FCF-relaxation, where $\hat{A}$ is the "lumped" matrix of strong connections computed from A.

	$\theta = 0$			$\theta = \pi/6$			$\theta = \pi/4$		
Grid size	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}
16×16	0.168	1.50	1.51	0.106	1.48	1.97	0.062	1.47	1.63
32×32	0.174	1.50	1.53	0.176	1.50	2.04	0.069	1.49	1.72
64×64	0.182	1.50	1.55	0.227	1.50	2.09	0.075	1.51	1.75
128×128	0.189	1.51	1.56	0.260	1.51	2.14	0.075	1.51	1.77

matrices lead to even higher three-grid complexities in the results to follow. Thus, we focus on the choice of $\eta = 0.65$ in the results below.

Interpolation truncation: To attenuate the higher complexities observed in Tables 7 and 8, we employ interpolation truncation, which is used successfully in several AMG settings[28–30]. In our approach, we first use SPAI (with a sparsity pattern of $\hat{A}_{FC} + \hat{A}_{FF}\hat{A}_{FC}$) to compute $W \approx \hat{A}_{FF}^{-1}\hat{A}_{FC}$. Then, we sweep row-wise through W, dropping entries that are less than a factor of ζ of the largest entry (by absolute value) in the row, to yield $\widehat{W}$. As a final step, we compute matrix S to match interpolation to the relaxed vector, $\boldsymbol{z}$, so that $\boldsymbol{z}_F = S\widehat{W}\boldsymbol{z}_C$. Table 9 shows results using $\zeta = 0.2$. For the cases of $\theta = 0$ and $\pi/4$, we see that this truncation has no real effect in comparison with results in Table 7. This is easily understood from the nature of strong connections in these matrices, with only two strong connections per row, so there are only two interpolation weights in a typical row, and these weights are roughly equal in size. In such cases, no effects of this truncation are expected. For $\theta = \pi/6$, we see that this truncation leads to slight improvements in operator complexities, with small effects on convergence factors. While the savings here may be minimal, we show below that interpolation truncation is an important tool in other cases, such as the isotropic Poisson problem in Section 5.1. We also note that further increasing the truncation parameter to $\zeta = 0.25$ starts to show significantly degraded performance for $\theta = \pi/6$, when "too many" connections in interpolation are truncated.

Tables 10 and 11 compare no-interpolation truncation, equivalent to $\zeta = 0$, to that of truncation with $\zeta = 0.2$ for three-level cycles for these problems. As expected, adding more levels to the hierarchy increases the grid and operator complexities. Using two-level geometric semi-coarsening-by-threes as a reference, we expect to see $C_{\text{grid}} \approx 1 + \frac{1}{3} + \frac{1}{9} \approx 1.44$, which we do in all cases (with a slight increase for $\theta = \pi/6$, undoubtedly due to increased density of the coarse-grid operators). We note that convergence does degrade going from two-grid to three-grid cycles, particularly for V-cycles (with convergence factors denoted by ρ_V), but also for W-cycles (with convergence factors denoted by ρ_W). Again, the effects of truncation are minimal for $\theta = 0$ and $\pi/4$, with some noticeable, but small, changes for $\pi/6$.

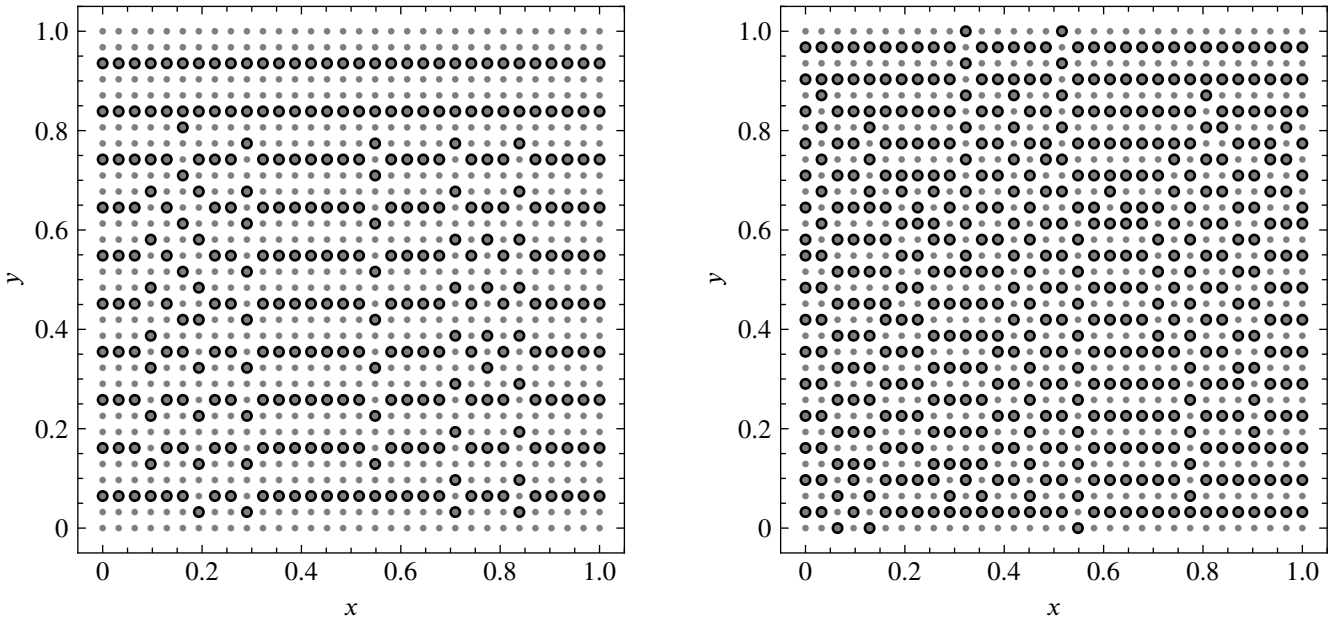

FIGURE 1 Algebraic coarse-fine partitionings for the FE discretization of anisotropic-diffusion with $\theta = 0$ on uniform 32×32 grid. At left, partitioning with $\eta = 0.65$. At right, partitioning with $\eta = 0.75$. Fine-grid DoFs (points in Ω) are denoted by filled grey dots; those that are in C are marked with black circles.

TABLE 9 Two-level AMGr convergence factors and corresponding complexities for anisotropic FE discretization using simulated annealing coarsening with $\eta = 0.65$. F-nodes are interpolated using the SPAI approximation to $\hat{A}_{FF}^{-1}\hat{A}_{FC}$, postprocessed to exactly interpolate a relaxed vector, and the SPAI approximations to $\hat{A}_{FF}^{-1}$ and $\hat{A}_{CC}^{-1}$ are used for FCF-relaxation, where $\hat{A}$ is the "lumped" matrix of strong connections computed from A. **Interpolation truncation with $\zeta = 0.2$ is used.**

	$\theta = 0$			$\theta = \pi/6$			$\theta = \pi/4$		
Grid size	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}
16×16	0.219	1.32	1.32	0.107	1.35	1.68	0.116	1.30	1.40
32×32	0.235	1.32	1.33	0.194	1.35	1.77	0.114	1.32	1.49
64×64	0.234	1.34	1.37	0.364	1.36	1.88	0.121	1.33	1.54
128×128	0.231	1.34	1.39	0.408	1.36	1.93	0.133	1.34	1.57

4.2 | The Generalized AMGr algorithm

Before presenting more extensive numerical results, we summarize the outcome of the experiments above in algorithmic form. Algorithm 4 presents the AMGr setup algorithm: we compute the lumped matrix, $\hat{A}$, after finding strong connections, construct the F-C partitioning based on $\hat{A}$, then form interpolation and the Galerkin coarse-grid operator. In addition, we compute SPAI approximations to the inverses of $\hat{A}_{FF}$ and $\hat{A}_{CC}$, for use in relaxation, using $\mathcal{S}(A)$ to denote the sparsity pattern of matrix A. As in the original AMGr paper, we use weighted relaxatio, with optimal weights for single-step relaxation on the F and C subproblems, computed based on eigenvalue estimates for the "preconditioned" matrices, $D_{FF}^{\text{inv}}A_{FF}$ and $D_{CC}^{\text{inv}}A_{CC}$. For the coarse-grid problem, the computed eigenvalues are very close to one in all cases, and replacing the weighted relaxation with unweighted relaxation (approximating $\sigma_C = 1$) has little effect on convergence. For the fine-grid relaxation, we find more variation in λ_{min} and λ_{max}; we explore more practical alternatives to determining relaxation weights in Section 5.4, although note that it may also be possible to choose better weights row-wise (e.g., by using weighting similar to the ℓ^1-Jacobi relaxation method[58] or the matrix S determined for interpolation).

TABLE 10 Three-level AMGr convergence factors and corresponding complexities for anisotropic FE discretization using simulated annealing coarsening with $\eta = 0.65$. F-nodes are interpolated using the SPAI approximation to $\hat{A}_{FF}^{-1}\hat{A}_{FC}$, postprocessed to exactly interpolate a relaxed vector, and the SPAI approximations to $\hat{A}_{FF}^{-1}$ and $\hat{A}_{CC}^{-1}$ are used for FCF-relaxation, where $\hat{A}$ is the "lumped" matrix of strong connections computed from A.

	$\theta = 0$				$\theta = \pi/6$				$\theta = \pi/4$			
Grid size	ρ_V	ρ_W	C_{grid}	C_{op}	ρ_V	ρ_W	C_{grid}	C_{op}	ρ_V	ρ_W	C_{grid}	C_{op}
16×16	0.238	0.221	1.40	1.36	0.169	0.116	1.46	1.92	0.158	0.116	1.41	1.57
32×32	0.240	0.235	1.42	1.41	0.393	0.248	1.47	2.18	0.320	0.114	1.43	1.70
64×64	0.243	0.234	1.45	1.49	0.537	0.394	1.49	2.36	0.448	0.224	1.45	1.79
128×128	0.267	0.231	1.46	1.52	0.656	0.502	1.50	2.44	0.541	0.310	1.47	1.88

TABLE 11 Three-level AMGr convergence factors and corresponding complexities for anisotropic FE discretization using simulated annealing coarsening with $\eta = 0.65$. F-nodes are interpolated using the SPAI approximation to $\hat{A}_{FF}^{-1}\hat{A}_{FC}$, postprocessed to exactly interpolate a relaxed vector, and the SPAI approximations to $\hat{A}_{FF}^{-1}$ and $\hat{A}_{CC}^{-1}$ are used for FCF-relaxation, where $\hat{A}$ is the "lumped" matrix of strong connections computed from A. **Interpolation truncation with $\zeta = 0.2$ is employed.**

	$\theta = 0$				$\theta = \pi/6$				$\theta = \pi/4$			
Grid size	ρ_V	ρ_W	C_{grid}	C_{op}	ρ_V	ρ_W	C_{grid}	C_{op}	ρ_V	ρ_W	C_{grid}	C_{op}
16×16	0.238	0.220	1.40	1.36	0.175	0.113	1.46	1.85	0.153	0.116	1.41	1.57
32×32	0.248	0.235	1.42	1.41	0.359	0.231	1.47	2.06	0.323	0.114	1.44	1.70
64×64	0.243	0.234	1.45	1.49	0.556	0.411	1.49	2.24	0.437	0.215	1.46	1.79
128×128	0.262	0.231	1.46	1.52	0.670	0.542	1.50	2.31	0.534	0.298	1.47	1.87

Algorithm 4 Generalized AMGr Setup Phase

function GEN-AMGR-SETUP(A, b)
  $\hat{A} \leftarrow$ lumped approximation to A after removing weak connections
  $C, F \leftarrow$ partitioning based on $\hat{A}$
  $\hat{A}_{FF}, \hat{A}_{FC}, \hat{A}_{CC} \leftarrow$ extract submatrices of $\hat{A}$ based on F and C
  $P \leftarrow$ INTERPOLATION($\hat{A}, \hat{A}_{FF}, \hat{A}_{FC}, F, C$)
  $A_C \leftarrow P^T A P$
  $D_{FF}^{\text{inv}} \leftarrow$ SPAI($\hat{A}_{FF}, I_{FF}, \mathcal{S}\left(\hat{A}_{FF}\right)$)
  $\lambda_{\min}, \lambda_{\max} \leftarrow$ minimum and maximum eigenvalues of $D_{FF}^{\text{inv}} A_{FF}$
  $\sigma_F \leftarrow 2/(\lambda_{\min} + \lambda_{\max})$
  $D_{CC}^{\text{inv}} \leftarrow$ SPAI($\hat{A}_{CC}, I_{CC}, \mathcal{S}\left(\hat{A}_{CC}\right)$)
  $\lambda_{\min}, \lambda_{\max} \leftarrow$ minimum and maximum eigenvalues of $D_{CC}^{\text{inv}} A_{CC}$
  $\sigma_C \leftarrow 2/(\lambda_{\min} + \lambda_{\max})$
  return $F, C, P, A_C, D_{FF}^{\text{inv}}, \sigma_F, D_{CC}^{\text{inv}}, \sigma_C$
end function

The interpolation operator is computed following Algorithm 5. First, the nonzero pattern is determined for interpolation, followed by a SPAI approximation to $\hat{A}_{FF}^{-1}\hat{A}_{FC}$ for this pattern. Small entries may be truncated in W at this stage in order to reduce complexity of the resulting cycle. After truncation, we rescale interpolation row-wise, either using constant scaling, as in Algorithm 6, or the improved iteration scaling, as in Algorithm 7.

Finally, we present the two-level AMGr solution phase in Algorithm 8. This includes either F- or FCF-relaxation both before and after the coarse-grid correction phase, as well as a standard Galerkin coarse-grid correction. While we present the algorithm without implementation details, we note that the algorithms here can be implemented in either the "natural" ordering of matrix

Algorithm 5 Interpolation Operator

function INTERPOLATION($\hat{A}$, $\hat{A}_{FF}$, $\hat{A}_{FC}$, F, C)
  $\mathcal{Z} \leftarrow \mathcal{S}\left(\hat{A}_{FC} + \hat{A}_{FF}\hat{A}_{FC}\right)$
  $W \leftarrow$ SPAI($\hat{A}_{FF}, \hat{A}_{FC}, \mathcal{Z}$)
  $W \leftarrow W$ after truncating small entries
  if improved iteration scaling to be used **then**
    $W \leftarrow$ IMPROVED-ITERATION-SCALING($\hat{A}, W, F, C$)
  else
    $W \leftarrow$ CONSTANT-SCALING(W, F, C)
  end if
  $P \leftarrow \begin{bmatrix} W \\ I \end{bmatrix}$
  return P
end function

Algorithm 6 Constant Scaling

function CONSTANT-SCALING(W, F, C)
  $\boldsymbol{s} \leftarrow \mathbf{1}_F/(W\mathbf{1}_C)$ ▷ Componentwise division
  $S \leftarrow$ matrix with $\boldsymbol{s}$ on diagonal and elsewhere 0
  $W \leftarrow SW$
  return W
end function

Algorithm 7 Improved Iteration Scaling

function IMPROVED-ITERATION-SCALING($\hat{A}, W, F, C$)
  $n_{\text{wj}} \leftarrow 5, \omega \leftarrow 2/3, D \leftarrow$ diagonal of $\hat{A}$
  $\boldsymbol{z} \leftarrow \mathbf{1}$
  for $i \leftarrow 1, \ldots, n_{\text{wj}}$ **do**
    $\boldsymbol{z} \leftarrow (I - \omega D^{-1}\hat{A})\boldsymbol{z}$
  end for
  $\boldsymbol{s} \leftarrow \boldsymbol{z}_F/(W\boldsymbol{z}_C)$ ▷ Componentwise division
  $S \leftarrow$ matrix with $\boldsymbol{s}$ on diagonal and elsewhere 0
  $W \leftarrow SW$
  return W
end function

A, or in the "permuted" ordering given in (3). In many ways, it is simpler to implement the algorithm after permuting A into its F-C ordering.

5 | RESULTS

While the algorithms given above were derived by focusing on performance for finite-element discretizations of anisotropic diffusion equations on uniform grids, we emphasize in this section that this methodology appears to have much wider applicability. Here, we first evaluate the approach on isotropic diffusion equations, on both structured and unstructured grids. Furthermore, we consider results for a classic "four-quadrant" problem, with piecewise constant diffusion and reaction coefficients on a uniform grid, and for constant-coefficient anisotropic diffusion on an unstructured grid. In all results before Section 5.4, we use the simulated annealing coarsening algorithm described above with $\eta = 0.65$.

Algorithm 8 Generalized AMGr Solution Phase

function TWO-LEVEL($A, \boldsymbol{b}, \boldsymbol{u}, F, C, P, A_C, D_{FF}^{\text{inv}}, \sigma_F, D_{CC}^{\text{inv}}, \sigma_C$)
  $\boldsymbol{u} \leftarrow F$-relaxation on $A\boldsymbol{u} = \boldsymbol{b}$
  if C-relaxation to be used **then**
    $\boldsymbol{u} \leftarrow C$-relaxation on $A\boldsymbol{u} = \boldsymbol{b}$
    $\boldsymbol{u} \leftarrow F$-relaxation on $A\boldsymbol{u} = \boldsymbol{b}$
  end if
  $\boldsymbol{r}_C \leftarrow P^T(\boldsymbol{b} - A\boldsymbol{u})$
  $\boldsymbol{e}_C \leftarrow$ solution of $A_C\boldsymbol{e}_C = \boldsymbol{r}_C$
  $\boldsymbol{u} \leftarrow \boldsymbol{u} + P\boldsymbol{e}_C$
  $\boldsymbol{u} \leftarrow F$-relaxation on $A\boldsymbol{u} = \boldsymbol{b}$
  if C-relaxation to be used **then**
    $\boldsymbol{u} \leftarrow C$-relaxation on $A\boldsymbol{u} = \boldsymbol{b}$
    $\boldsymbol{u} \leftarrow F$-relaxation on $A\boldsymbol{u} = \boldsymbol{b}$
  end if
  return $\boldsymbol{u}$
end function

TABLE 12 AMGr convergence factors and complexities for **isotropic FE discretization on structured grids**. Results for classical (diagonal D_{FF}) AMGr appear in the first block column, followed by those for the two-level and three-level SPAI-based algorithm in Section 4.2 in following block columns, using $\zeta = 0$.

	Classical Two-level cycle			Two-level cycle			Three-level cycles			
Grid size	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}	ρ_V	ρ_W	C_{grid}	C_{op}
16×16	0.365	1.36	1.71	0.041	1.36	2.22	0.097	0.044	1.49	2.46
32×32	0.375	1.38	1.80	0.041	1.38	2.64	0.094	0.042	1.53	3.07
64×64	0.373	1.40	1.86	0.039	1.40	2.86	0.102	0.042	1.55	3.37
128×128	0.367	1.41	1.90	0.040	1.41	2.97	0.120	0.042	1.57	3.54

5.1 | Isotropic Poisson problem

In this section, we consider the isotropic diffusion equation $-\Delta u = f$, with Dirichlet boundary conditions, first on uniform meshes of the unit square domain. As a benchmark, the first block column in Table 12 presents convergence for the classical AMGr algorithm using a diagonal approximation, D_{FF}, to A_{FF} in both relaxation and interpolation with only F-relaxation. We note that using $\eta = 0.65$ already yields a positive effect on convergence; using $\eta = 0.56$ (as considered in past work) leads to convergence factors around 0.7, instead of 0.37. In either case, while the convergence factors are bounded away from unity independently of grid size, the convergence is suboptimal for AMG on the model Poisson equation on a uniform grid. The remaining columns of Table 12 present results for the algorithm of Section 4.2, demonstrating substantial improvement in two-grid convergence and reasonable three-level convergence. We also note that the two-level generalized AMGr algorithm using F-relaxation in place of FCF-relaxation also offers reasonable convergence factors of about 0.1. Here, we see that while the grid complexities for these cycles are relatively reasonable, the operator complexities are high, above 3.0 for most of the three-level cycles.

To reduce the computational complexities, we truncate the smaller elements in the interpolation operator as discussed above. A critical question in using interpolation truncation is the choice of the value of parameter ζ. The left plot of Figure 2 shows the effects of varying this parameter for the 32×32 uniform grid. For small values of ζ, we observe large complexities, but also excellent two-level convergence factors. As ζ increases past 0.2, so do the convergence factors, yet the complexity continues drop. If we were solely concerned with convergence, we might conclude that this is the optimal value of ζ, since it yields the lowest complexity while retaining the best-possible convergence factor. However, to better balance cost vs. complexity, we prefer to take $\zeta = 0.25$, where we approximately minimize the two-level complexity, while still retaining an acceptable convergence

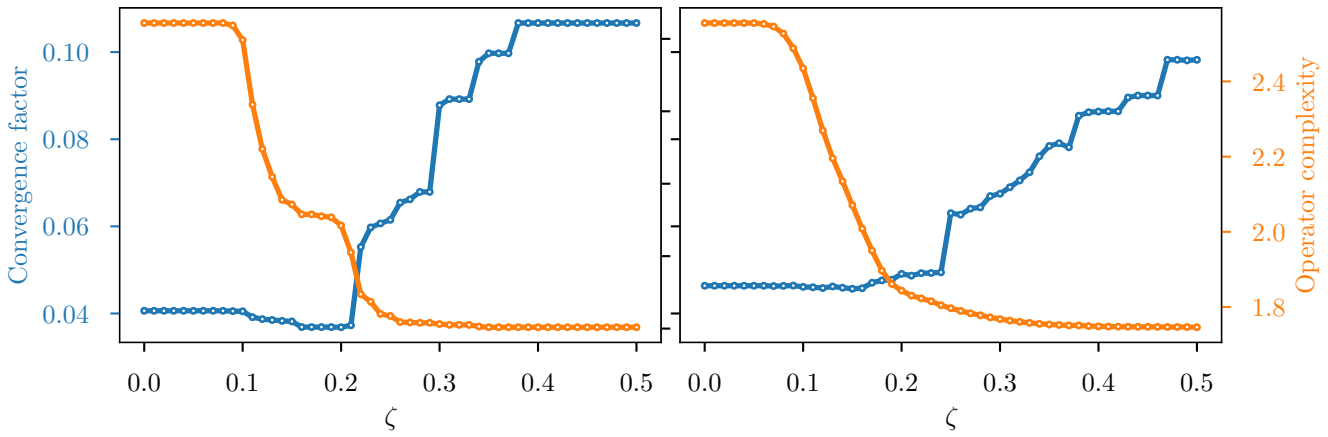

FIGURE 2 Trade-off between two-level convergence factor and operator complexities as a function of ζ, for isotropic Poisson on a uniform 32×32 grid (at left) and on the unstructured triangulation with 1433 DoFs (at right).

TABLE 13 AMGr convergence factors and complexities for isotropic FE discretization on structured grids, using the two-level and three-level SPAI-based algorithm in Section 4.2, **with** $\zeta = 0.25$.

	Two-level cycle			Three-level cycles			
Grid size	ρ	C_{grid}	C_{op}	ρ_V	ρ_W	C_{grid}	C_{op}
16×16	0.053	1.36	1.73	0.099	0.058	1.49	1.90
32×32	0.061	1.38	1.84	0.140	0.071	1.52	2.07
64×64	0.065	1.40	1.91	0.151	0.074	1.55	2.20
128×128	0.069	1.41	1.95	0.157	0.079	1.56	2.26

factor. Table 13 shows two- and three-grid performance as we vary grid size with $\zeta = 0.25$. We see substantial improvements in complexity, with two-level complexities now similar to those of the classical AMGr algorithm in Table 12, and three-level grid complexities now about 2.2, instead of over 3.0. At the same time, excellent two-level convergence factors are maintained, and there is only a slight impact on three-level convergence factors.

An important consideration for algebraic multigrid methods is whether or not they retain their performance as we transition from structured to unstructured grids. Hence, our next problem considers the same isotropic diffusion operator, but discretized using piecewise linear finite elements on unstructured triangulations of the square domain, $[-1, 1]^2$. We construct grids by starting from an unstructured grid, performing several steps of uniform refinement, then smoothing the resulting grids. Here, we consider three levels of refinement, generating meshes with 1433, 5617, and 22 241 DoFs. We again study the effects of varying the truncation parameter, ζ, at right of Figure 2, and conclude that taking $\zeta = 0.25$ again gives a good trade-off between convergence and complexity. Table 14 shows the resulting two- and three-grid convergence factors and operator complexities for the new AMGr algorithm applied to these problems. While the convergence factors are somewhat larger than those for the uniform-grid discretization, they remain acceptable for AMG convergence for an isotropic diffusion operator. Furthermore, we see the efficacy of interpolation truncation in reducing the operator complexity while maintaining acceptable convergence factors.

5.2 | Four-quadrant problems

Next, we consider a family of two-dimensional anisotropic diffusion problems by adding a reaction term to Equation (7), giving

$$-\nabla \cdot \boldsymbol{K}(x, y)\nabla u(x, y) + c(x, y)u(x, y) = b(x, y) \tag{10}$$

TABLE 14 AMGr convergence factors and complexities for isotropic FE discretization on **unstructured grids**, using the two-level and three-level SPAI-based algorithm in Section 4.2. Results in the left-most block column show two-level results with no interpolation truncation ($\zeta = 0$), while the other block columns show results with $\zeta = 0.25$.

	without truncation			with interpolation truncation						
	Two-level cycle			Two-level cycle			Three-level cycles			
#DoF	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}	ρ_V	ρ_W	C_{grid}	C_{op}
1433	0.046	1.36	2.56	0.063	1.36	1.80	0.155	0.064	1.50	2.17
5617	0.135	1.36	2.53	0.132	1.36	1.75	0.167	0.127	1.50	2.17
22241	0.167	1.37	2.49	0.167	1.37	1.78	0.322	0.186	1.52	2.22

in the domain $[0, 1] \times [0, 1]$ with Dirichlet boundary conditions. The tensor coefficient is chosen as $\boldsymbol{K}(x, y) = QHQ^T$, where $Q = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$, and $H = \begin{bmatrix} 1 & 0 \\ 0 & \delta \end{bmatrix}$, where θ specifies the direction of anisotropy in the problem and δ specifies its strength. We note that the convention for H used here is different than that in Equation (8), to make the problems consistent with those used in other works. We partition the domain $[0, 1] \times [0, 1]$ into four equal quadrants and consider constant values of θ, δ and c within each quadrant, but with different values in different quadrants of the domain. The four-quadrant problem is common in AMG literature, and we consider three different problems within this class, with coefficient values shown below in Figure 3. Problem 1 is similar to the problem in Chapter 8 in the book by Briggs, Henson, and McCormick[59], with no reaction term, large contrasts in the anisotropy strength, and non-grid-aligned diffusion in just one quadrant. Problem 2 is the 2D-4Reg problem from Brannick and Falgout[60], with a large reaction coefficient in one quadrant, but a small contrast in anisotropy strength and only grid-aligned anisotropy. Finally, Problem 3 is constructed to provide a more significant challenge, including a large reaction coefficient in one quadrant, a large contrast in anisotropy strength, and anisotropy directions in two quadrants that are neither aligned with the grid nor with the grid diagonal.

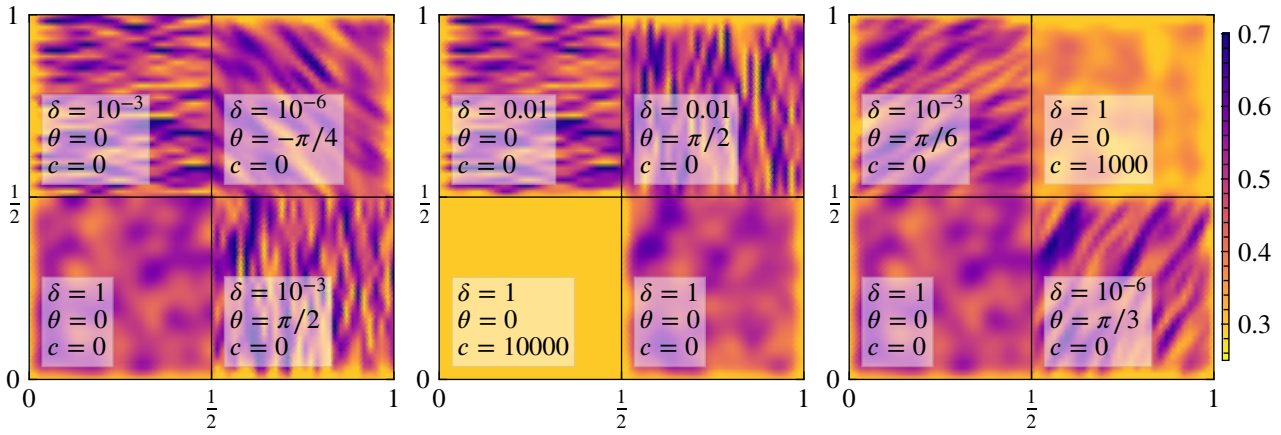

FIGURE 3 Visualization of the strong connections for the four-quadrant problems; Problem 1 (left), Problem 2 (middle), and Problem 3 (right).

Two-level and three-level AMGr performance for these problems is shown in Tables 15 and 16, respectively. We note that the two-level performance for Problems 1 and 2 is generally good, both in terms of convergence factor and complexity, while Problem 3 is clearly a harder problem. Indeed, both V- and W-cycle convergence continues to perform well for Problem 2 in the three-level results in Table 16, with convergence outperforming that reported for the 65×65 grid in Table 4.2 in Brannick and Falgout[60], with comparable grid and operator complexities to the compatible relaxation AMG solver proposed there, and much better complexities than those reported there for BoomerAMG[5]. Problem 3 is clearly more taxing for AMG, yet the proposed

TABLE 15 Two-level AMGr convergence factors and complexities for the **four-quadrant problem** using the SPAI-based algorithm in Section 4.2, with $\zeta = 0.25$.

Grid size	Problem 1			Problem 2			Problem 3		
	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}	ρ	C_{grid}	C_{op}
17×17	0.388	1.35	1.49	0.602	1.26	1.32	0.062	1.25	1.38
33×33	0.433	1.34	1.51	0.581	1.26	1.37	0.266	1.33	1.62
65×65	0.413	1.36	1.56	0.595	1.27	1.41	0.491	1.39	1.78
129×129	0.420	1.36	1.57	0.599	1.36	1.60	0.692	1.39	1.80

TABLE 16 Three-level AMGr convergence factors and complexities for the **four-quadrant problem** using the SPAI-based algorithm in Section 4.2, with $\zeta = 0.25$.

Grid size	Problem 1				Problem 2				Problem 3			
	ρ_V	ρ_W	C_{grid}	C_{op}	ρ_V	ρ_W	C_{grid}	C_{op}	ρ_V	ρ_W	C_{grid}	C_{op}
17×17	0.404	0.390	1.48	1.65	0.608	0.602	1.34	1.39	0.095	0.064	1.32	1.44
33×33	0.474	0.440	1.46	1.68	0.584	0.581	1.35	1.48	0.314	0.273	1.43	1.78
65×65	0.523	0.422	1.49	1.77	0.598	0.595	1.36	1.54	0.539	0.499	1.52	2.05
129×129	0.599	0.433	1.49	1.80	0.604	0.600	1.46	1.75	0.740	0.702	1.54	2.12

TABLE 17 Two- and three-level AMGr convergence factors and complexities for the **anisotropic diffusion problem on unstructured meshes** using the SPAI-based algorithm in Section 4.2, with $\zeta = 0.25$.

#DoF	Two-level cycle			Three-level cycles			
	ρ	C_{grid}	C_{op}	ρ_V	ρ_W	C_{grid}	C_{op}
798	0.516	1.35	1.64	0.586	0.534	1.49	1.95
3109	0.607	1.37	1.64	0.676	0.621	1.51	1.99
12273	0.601	1.37	1.65	0.707	0.639	1.51	2.01

generalized AMGr approach offers acceptable convergence in all cases. Whether further improvement to these results is possible (or the performance degradation with grid size can be attenuated using Krylov acceleration) is left for future work.

5.3 | Unstructured anisotropic diffusion

Next, we consider the anisotropic diffusion problem in (10) with Dirichlet boundary conditions, $c = 0$, $\theta = \pi/3$, and $\delta = 0.01$, on an unstructured triangulation of the unit square taken from Brannick and Falgout[60], where the problem is labeled as 2D-M2-RLap. As in Table 4.2 from Brannick and Falgout[60], we consider three refinements of the unstructured mesh for this problem, yielding discretized problems with 798, 3109, and 12 273 DoFs, respectively. The mesh containing 798 DoFs (and its coarsening using $\eta = 0.65$) is shown in Figure 4. Table 17 presents two- and three-level convergence results for the generalized AMGr algorithm applied to this problem. For comparison, we note that Table 4.2 of Brannick and Falgout[60] reports higher convergence factors (up to 0.95 on the finest grid) for CR-AMG applied to this problem, but at lower grid and operator complexities. Compared to the BoomerAMG results presented in the same table, we see comparable convergence (0.57 for the finest grid) at lower complexity (2.6 at the finest grid, albeit for a multilevel cycle, not a three-level cycle). Compared to classical AMGr applied to this problem, as given in Table 6 of Zaman et al.[19], we see substantial improvement in convergence factors (compared to values of 0.8–0.9 on the finest grid) and lower complexities in these results.

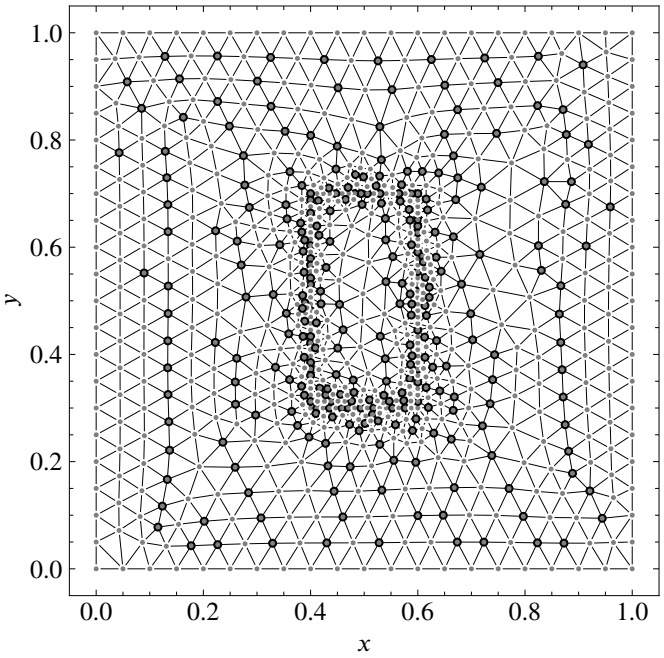

FIGURE 4 Unstructured triangulation containing 798 points from Brannick and Falgout[60], and its partitioning using $\eta = 0.65$. Fine-grid DoFs (points in Ω) are denoted by filled grey dots; those that are in C (282 points) are marked with black circles.

5.4 | Multilevel Results

Two major obstacles remain in Algorithm 4 for transitioning from the two- and three-level cycles studied above to standard multilevel cycles. First of all, we have (until now) focused on the use of simulated annealing for determining the partitioning of A and $\hat{A}$ into the F and C sets. While this is very effective, it is also very costly, as many SA steps are required to generate near-optimal partitionings using this algorithm. Thus, we switch here to using the greedy coarsening algorithm of MacLachlan and Saad[16], which is much more efficient, but generates poorer-quality partitions. To compensate, we investigate the effect of the diagonal dominance parameter, η, on the complexities and convergence of the resulting multilevel hierarchies, in order to attenuate some of the complexity growth that we observe in the initial results.

The second major obstacle is the calculation of extremal eigenvalues in Lines 8 and 11 of Algorithm 4, which has additional heavy computational cost. To eliminate this, we replace the optimal calculation of σ_F and σ_C with a common heuristic estimate of the optimal regularization parameter. Knowing that A_{FF} and A_{CC} are both positive-definite matrices, we expect that D_{FF}^{inv} and D_{CC}^{inv} are as well. If this is the case, the spectra of $D_{FF}^{\text{inv}}A_{FF}$ and $D_{CC}^{\text{inv}}A_{CC}$ are guaranteed to be contained in the intervals from 0 to their largest eigenvalues, which can be estimated by their maximum absolute row sums (using Geršgorin's theorem). While the F-relaxation originally used in AMGr targets an optimal reduction over all modes by estimating both ends of the spectrum of $D_{FF}^{\text{inv}}A_{FF}$, we propose a simpler heuristic of choosing σ_F and σ_C to be $3/2$ divided by the maximum absolute row sum of $D_{FF}^{\text{inv}}A_{FF}$ and $D_{CC}^{\text{inv}}A_{CC}$, respectively. The choice of weight $3/2$ in this heuristic reflects the expectation that these matrices are well-conditioned, so we need not use a weight as large as 2 (which would be optimal if we estimate the smallest eigenvalues as 0), but that they are far from perfectly conditioned, so the weight should be larger than 1. Numerical tests confirm that using weight $3/2$ is a good compromise — in some cases, some improvements are possible with larger weights, but this leads to greatly degraded performance in some cases as well.

As a comparison with the final three-level results in Table 11. Table 18 shows convergence factors, complexities, and number of levels in the multigrid hierarchies (n_l) for the multilevel algorithm. These results show notable degradation in both operator and grid complexities, due to the use of greedy coarsening with $\eta = 0.65$ in contrast with the simulated annealing coarsening used in the previous results. Nonetheless, we observe excellent W-cycle convergence factors in all cases (outperforming the earlier results for $\theta = 0$ and $\theta = \pi/6$), and consistent V-cycle convergence factors. In experiments not reported here, we compared convergence to the case of using exact eigenvalues and found little difference in convergence overall. Notably, when using the

TABLE 18 Multi-level AMGr convergence factors and corresponding complexities for anisotropic FE discretization using **greedy coarsening** with $\eta = 0.65$. F-nodes are interpolated using the SPAI approximation to $\hat{A}_{FF}^{-1}\hat{A}_{FC}$, postprocessed to exactly interpolate a relaxed vector, and the SPAI approximations to $\hat{A}_{FF}^{-1}$ and $\hat{A}_{CC}^{-1}$ are used for FCF-relaxation, where $\hat{A}$ is the "lumped" matrix of strong connections computed from A. Interpolation truncation with $\zeta = 0.2$ is employed. **Estimates of the eigenvalues are used in relaxation.**

	$\theta = 0$					$\theta = \pi/6$					$\theta = \pi/4$				
Grid size	ρ_V	ρ_W	C_{grid}	C_{op}	n_l	ρ_V	ρ_W	C_{grid}	C_{op}	n_l	ρ_V	ρ_W	C_{grid}	C_{op}	n_l
32×32	0.205	0.186	1.78	1.73	4	0.312	0.136	1.76	2.24	4	0.383	0.170	1.74	2.11	4
64×64	0.230	0.187	1.84	1.81	6	0.515	0.139	1.88	2.50	6	0.460	0.149	1.87	2.45	6
128×128	0.232	0.188	1.91	1.88	7	0.641	0.163	1.93	2.64	8	0.668	0.285	1.92	2.61	8
256×256	0.242	0.186	1.95	1.93	8	0.722	0.180	1.96	2.72	10	0.740	0.346	1.96	2.72	10

TABLE 19 Multi-level AMGr convergence factors and complexities for the **four-quadrant problem** using the SPAI-based algorithm in Section 4.2, with $\zeta = 0.25$. **Greedy coarsening and the heuristic eigenvalue estimates are used.**

	Problem 1					Problem 2					Problem 3				
Grid size	ρ_V	ρ_W	C_{grid}	C_{op}	n_l	ρ_V	ρ_W	C_{grid}	C_{op}	n_l	ρ_V	ρ_W	C_{grid}	C_{op}	n_l
33×33	0.541	0.412	1.77	2.08	4	0.429	0.334	1.55	1.73	4	0.202	0.097	1.66	2.17	4
65×65	0.602	0.419	1.87	2.30	6	0.435	0.263	1.65	1.92	6	0.414	0.128	1.84	2.59	6
129×129	0.698	0.421	1.93	2.43	8	0.323	0.256	1.83	2.34	8	0.621	0.151	1.92	2.82	8
257×257	0.783	0.420	1.97	2.49	11	0.367	0.258	1.95	2.63	11	0.769	0.210	1.96	2.95	10

multigrid cycles as preconditioners for conjugate gradient, using the heuristic choice incurs at most 3 additional iterations over using cycles based on the exact eigenvalue computation.

Similar results are shown in Table 19 for the four-quadrant problems from Section 5.2, for comparison with Table 16. Again, we note that the complexities are much higher than those reported earlier using the simulated annealing coarsening algorithm, but that this added complexity pays off in improved multilevel convergence. For Problem 2, we again compare to the results presented in Table 4.2 by Brannick and Falgout[60], and see that this coarsening achieves comparable complexities to those reported there for 33×33 and 65×65 grids, but much better convergence. Overall, we again see grid-independent W-cycle convergence for each problem, but growth in V-cycle convergence factors. When run as preconditioners for CG, we find that W-cycles lead to convergence in 5–10 iterations (more for Problem 1, fewer for Problems 2 and 3), and V-cycle convergence in up to 14 iterations (again, with Problem 1 requiring most, and Problem 3 requiring fewest).

Finally, we present results for the anisotropic diffusion problem on unstructured meshes considered in Section 5.3 and Table 17. Here, to explore the connection between the diagonal dominance parameter, η, and the resulting complexities and convergence factors, we consider $\eta = 0.65$ as before, along with $\eta = 0.60$ and $\eta = 0.56$. Table 20 shows that, as expected, complexities decrease and convergence factors generally increase as η gets smaller, but that significant improvements in complexity are possible by using smaller η without sacrificing substantial convergence. In particular, comparing results for $\eta = 0.56$, we observe modest increases in grid complexity in comparison with those in Table 17, possibly attributed to the increase from three-level to multi-level cycles. Comparing these results to that presented in Table 4.2 by Brannick and Falgout[60], we observe complexities better than those reported for BoomerAMG for these problems, albeit with slightly worse convergence, and slightly worse than those reported for compatible relaxation, but with better convergence.

6 | CONCLUSIONS AND FUTURE WORK

Reduction-based AMG methods have been proposed and studied in many settings over the past 15 years, building effective solvers that can be more closely related to AMG convergence theory than many other heuristic methods. In this paper, we aim to

TABLE 20 Multi-level AMGr convergence factors and complexities for **anisotropic diffusion problem on unstructured meshes** using the SPAI-based algorithm in Section 4.2, with $\zeta = 0.25$. **Greedy coarsening, for three values of η, and the heuristic eigenvalue estimates are used.**

	$\eta = 0.56$					$\eta = 0.60$					$\eta = 0.65$				
#DoF	ρ_V	ρ_W	C_{grid}	C_{op}	n_l	ρ_V	ρ_W	C_{grid}	C_{op}	n_l	ρ_V	ρ_W	C_{grid}	C_{op}	n_l
798	0.612	0.560	1.54	2.06	4	0.601	0.529	1.66	2.37	4	0.537	0.467	1.76	2.56	4
3109	0.732	0.665	1.66	2.36	5	0.712	0.591	1.77	2.66	6	0.686	0.577	1.90	2.98	6
12273	0.777	0.653	1.71	2.51	7	0.762	0.630	1.83	2.87	8	0.742	0.576	1.98	3.33	9

improve the practical performance of AMGr approaches by targeting tools that can greatly improve performance for anisotropic diffusion equations. Through extensive numerical results, we show that the combination of using SPAI[27, 57] to approximate A_{FF}^{-1} along with tools to control sparsity leads to effective solvers for both anisotropic and isotropic diffusion operators on structured and unstructured grids. In our view, this work points to weaknesses in the existing theory for AMGr-type methods, where approximations to $A_{FF}^{-1}A_{FC}$ have not been considered (to our knowledge) in any context, providing an opportunity for future work. This may also provide new insights into desirable properties of SPAI-like approximations in this context. Additionally, further experiments are needed to see how to adapt the methodology proposed here to an even broader set of challenging problems, including the indefinite Helmholtz equation and convection-dominated flows.

ACKNOWLEDGMENTS

The work of S.P.M. was partially supported by an NSERC Discovery Grant. This work does not have any conflicts of interest.

References

1. Brandt A, McCormick SF, and Ruge JW. Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations; 1982. Institute for Computational Studies, Colorado State University.

2. Stüben K. Algebraic multigrid (AMG): Experiences and comparisons. Applied Mathematics and Computation. 1983;**13**(3-4):419–451.

3. Brandt A, McCormick SF, and Ruge JW. Algebraic multigrid (AMG) for sparse matrix equations. In: Evans DJ, editor. Sparsity and Its Applications. Cambridge: Cambridge University Press; 1984. p. 257–284.

4. Ruge JW, and Stüben K. Algebraic multigrid (AMG). In: McCormick SF, editor. Multigrid Methods. vol. 3 of Frontiers in Applied Mathematics. Philadelphia, PA: SIAM; 1987. p. 73–130.

5. Henson VE, and Yang UM. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. Applied Numerical Mathematics. 2002;**41**:155–177.

6. Falgout RD, and Yang UM. Hypre: A library of high performance preconditioners. In: Sloot PMA, Tan CJK, Dongarra JJ, and Hoekstra AG, editors. Computational Science - ICCS 2002: International Conference, Amsterdam, The Netherlands, April 21-24, 2002. Proceedings, Part III. No. 2331 in Lecture Notes in Computer Science. Springer-Verlag; 2002. p. 632–641.

7. Bell N, Olson LN, and Schroder J. PyAMG: Algebraic multigrid solvers in python. Journal of Open Source Software. 2022;**7**(72):4142. Available from: https://doi.org/10.21105/joss.04142.

8. Napov A, and Notay Y. An algebraic multigrid method with guaranteed convergence rate. SIAM Journal on Scientific Computing. 2012;**34**(2):A1079–A1109. Available from: https://doi.org/10.1137/100818509.

9. Brannick J, Chen Y, Kraus J, and Zikatanov L. An algebraic multigrid method based on matching in graphs. In: Domain Decomposition Methods in Science and Engineering XX. vol. 91 of Lecture Notes in Computational Science and Engineering. Springer, Heidelberg; 2013. p. 143–150. Available from: https://doi.org/10.1007/978-3-642-35275-1_15.

10. Brannick J, Chen Y, Kraus J, and Zikatanov L. Algebraic multilevel preconditioners for the graph Laplacian based on matching in graphs. SIAM Journal on Numerical Analysis. 2013;**51**(3):1805–1827. Available from: https://doi.org/10.1137/120876083.

11. Napov A, and Notay Y. An efficient multigrid method for graph Laplacian systems. Electronic Transactions on Numerical Analysis. 2016;**45**:201–218.

12. Napov A, and Notay Y. An efficient multigrid method for graph Laplacian systems II: Robust aggregation. SIAM Journal on Scientific Computing. 2017;**39**(5):S379–S403. Available from: https://doi.org/10.1137/16M1071420.

13. Ries M, Trottenberg U, and Winter G. A note on MGR methods. Linear Algebra and its Applications. 1983;**49**:1–26.

14. MacLachlan S, Manteuffel T, and McCormick S. Adaptive reduction-based AMG. Numerical Linear Algebra with Applications. 2006;**13**:599–620.

15. Swarztrauber PN. The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle. SIAM Review. 1977;**19**(3):490–501.

16. MacLachlan S, and Saad Y. A greedy strategy for coarse-grid selection. SIAM Journal on Scientific Computing. 2007;**29**(5):1825–1853.

17. Gossler F, and Nabben R. On AMG methods with F-smoothing based on Chebyshev polynomials and their relation to AMGr. Electronic Transactions on Numerical Analysis. 2016;**45**:146–159.

18. Brannick J, Frommer A, Kahl K, MacLachlan S, and Zikatanov L. Adaptive reduction-based multigrid for nearly singular and highly disordered physical systems. Electronic Transactions on Numerical Analysis. 2010;**37**:276–295.

19. Zaman T, MacLachlan SP, Olson LN, and West M. Coarse-grid selection using simulated annealing. arXiv preprint arXiv:210513280. 2021;.

20. Taghibakhshi A, MacLachlan S, Olson L, and West M. Optimization-based algebraic multigrid coarsening using reinforcement learning. In: Ranzato M, Beygelzimer A, Dauphin Y, Liang PS, and Vaughan JW, editors. Advances in Neural Information Processing Systems. vol. 34. Curran Associates, Inc.; 2021. p. 12129–12140.

21. MacLachlan S, and Saad Y. Greedy coarsening strategies for nonsymmetric problems. SIAM Journal on Scientific Computing. 2007;**29**(5):2115–2143.

22. Manteuffel TA, Ruge J, and Southworth BS. Nonsymmetric algebraic multigrid based on local approximate ideal restriction (ℓAIR). SIAM Journal on Scientific Computing. 2018;**40**(6):A4105–A4130.

23. Manteuffel TA, Münzenmaier S, Ruge J, and Southworth B. Nonsymmetric reduction-based algebraic multigrid. SIAM Journal on Scientific Computing. 2019;**41**(5):S242–S268.

24. Falgout RD, Friedhoff S, Kolev TV, MacLachlan SP, and Schroder JB. Parallel time integration with multigrid. SIAM Journal on Scientific Computing. 2014;**14**(1):951–952.

25. Kolotilina LY, and Yeremin AY. Factorized sparse approximate inverse preconditionings I. Theory. SIAM Journal on Matrix Analysis and Applications. 1993;**14**(1):45–58.

26. Benzi M, Meyer CD, and Tůma M. A sparse approximate inverse preconditioner for the conjugate gradient method. SIAM Journal on Scientific Computing. 1996;**17**(5):1135–1149.

27. Grote MJ, and Huckle T. Parallel preconditioning with sparse approximate inverses. SIAM Journal on Scientific Computing. 1997;**18**(3):838–853.

28. Stüben K. An introduction to algebraic multigrid. In: Trottenberg U, Oosterlee C, and Schüller A, editors. Multigrid. London: Academic Press; 2001. p. 413–528.

29. De Sterck H, Yang UM, and Heys JJ. Reducing complexity in parallel algebraic multigrid preconditioners. SIAM Journal on Matrix Analysis and Applications. 2006;**27**(4):1019–1039.

30. De Sterck H, Falgout RD, Nolting JW, and Yang UM. Distance-two interpolation for parallel algebraic multigrid. Numerical Linear Algebra with Applications. 2008;**15**(2-3):115–139.

31. Frederickson PO. Fast approximate inversion of large sparse linear systems. Lakehead University, Department of Mathematical Sciences; 1975.

32. Benson MW. Iterative solution of large scale linear systems (M.Sc. Thesis). Lakehead University. Thunder Bay, Canada; 1973.

33. Benson MW, and Frederickson PO. Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems. Utilitas Math. 1982;**22**:127–140.

34. Chow E, and Saad Y. Approximate inverse preconditioners via sparse-sparse iterations. SIAM Journal on Scientific Computing. 1998;**19**(3):995–1023.

35. Benson MW. Frequency domain behavior of a set of parallel multigrid smoothing operators. International Journal of Computer Mathematics. 1990;**36**(1-2):77–88.

36. Bröker O, Grote MJ, Mayer C, and Reusken A. Robust parallel smoothing for multigrid via sparse approximate inverses. SIAM Journal on Scientific Computing. 2001;**23**(4):1396–1417.

37. Bröker O, and Grote MJ. Sparse approximate inverse smoothers for geometric and algebraic multigrid. Applied Numerical Mathematics. 2002;**41**(1):61–80.

38. Bollhöfer M, and Mehrmann V. Algebraic multilevel methods and sparse approximate inverses. SIAM Journal on Matrix Analysis and Applications. 2002;**24**(1):191–218.

39. Bröker O. Parallel multigrid methods using sparse approximate inverses (Ph.D. thesis). ETH Zurich. Zurich, Switzerland; 2003.

40. Wang S, and de Sturler E. Multilevel sparse approximate inverse preconditioners for adaptive mesh refinement. Linear algebra and its applications. 2009;**431**(3-4):409–426.

41. Gravvanis GA, Filelis-Papadopoulos CK, and Matskanidis PI. Algebraic multigrid methods based on generic approximate inverse matrix techniques. Computer Modeling in Engineering & Sciences. 2014;**100**(4):323–345.

42. Filelis-Papadopoulos CK, and Gravvanis GA. Parallel multigrid algorithms based on generic approximate sparse inverses: an SMP approach. The Journal of Supercomputing. 2014;**67**(2):384–407.

43. Wagner C. On the algebraic construction of multilevel transfer operators. Computing. 2000;**65**(1):73–95.

44. Nägel A, Falgout RD, and Wittum G. Filtering algebraic multigrid and adaptive strategies. Computing and Visualization in Science. 2008;**11**(3):159–167.

45. Bolten M, Huckle TK, and Kravvaritis CD. Sparse matrix approximations for multigrid methods. Linear Algebra and its Applications. 2016;**502**:58–76.

46. Bollhöfer M. 2002. *Adapted Sparse Approximate Inverse Smoothers in Algebraic Multilevel Methods*. Preprint 759-2002. Institute of Mathematics, Technische Universität Berlin.

47. Meurant G. Numerical experiments with algebraic multilevel preconditioners. Electronic Transactions on Numerical Analysis. 2001;**12**:1–65.

48. Meurant G. A multilevel AINV preconditioner. Numerical Algorithms. 2002;**29**(1):107–129.

49. Benzi M, Cullum JK, and Tuma M. Robust approximate inverse preconditioning for the conjugate gradient method. SIAM Journal on Scientific Computing. 2000;**22**(4):1318–1332.

50. Brannick J, Brezina M, MacLachlan S, Manteuffel T, McCormick S, and Ruge J. An energy-based AMG coarsening strategy. Numerical Linear Algebra with Applications. 2006;**13**(2-3):133–148.

51. Olson LN, Schroder J, and Tuminaro RS. A new perspective on strength measures in algebraic multigrid. Numerical Linear Algebra with Applications. 2010;**17**(4):713–733.

52. Brandt A, Brannick J, Kahl K, and Livshits I. An algebraic distances measure of AMG strength of connection. arXiv preprint arXiv:11065990. 2011;.

53. Falgout RD, and Vassilevski PS. On generalizing the AMG framework. SIAM Journal on Numerical Analysis. 2004;**42**(4):1669–1693.

54. Mense C, and Nabben R. On algebraic multilevel methods for non-symmetric systems-convergence results. Electronic Transactions on Numerical Analysis. 2008;**30**:323–345.

55. Varga RS. Matrix Iterative Analysis. Springer Series in Computational Mathematics. Berlin: Springer; 2000. Second Edition.

56. Benzi M, and Tuma M. A comparative study of sparse approximate inverse preconditioners. Applied Numerical Mathematics. 1999;**30**(2-3):305–340.

57. Hawkins SC, and Chen K. An implicit wavelet sparse approximate inverse preconditioner. SIAM Journal on Scientific Computing. 2005;**27**(2):667–686.

58. Baker AH, Falgout RD, Kolev TV, and Yang UM. Multigrid smoothers for ultraparallel computing. SIAM Journal on Scientific Computing. 2011;**33**(5):2864–2887.

59. Briggs WL, Henson VE, and McCormick SF. A multigrid tutorial. SIAM; 2000.

60. Brannick JJ, and Falgout RD. Compatible relaxation and coarsening in algebraic multigrid. SIAM Journal on Scientific Computing. 2010;**32**(3):1393–1416.