LA-UR-23-27101
Accepted Manuscript

Constrained Local Approximate Ideal Restriction for Advection-Diffusion Problems

Ali, Ahsan
Brannick, James
Kahl, Karsten
Krzysik, Oliver
Schroder, Jacob
Southworth, Benjamin Scott

Provided by the author(s) and the Los Alamos National Laboratory (1930-01-01).

To be published in: SIAM Journal on Scientific Computing

DOI to publisher's version: 10.1137/23M1583442

Permalink to record:
https://permalink.lanl.gov/object/view?what=info:lanl-repo/lareport/LA-UR-23-27101

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

arXiv:2307.00229v2 [math.NA] 14 May 2024

CONSTRAINED LOCAL APPROXIMATE IDEAL RESTRICTION FOR ADVECTION-DIFFUSION PROBLEMS*

AHSAN ALI†, JAMES J. BRANNICK‡, KARSTEN KAHL§, OLIVER A. KRZYSIK¶, JACOB B. SCHRODER‖, AND BEN S. SOUTHWORTH#

Abstract. This paper focuses on developing a reduction-based algebraic multigrid (AMG) method that is suitable for solving general (non)symmetric linear systems and is naturally robust from pure advection to pure diffusion. Initial motivation comes from a new reduction-based algebraic multigrid approach, ℓAIR (local approximate ideal restriction), that was developed for solving advection-dominated problems. Though this new solver is very effective in the advection dominated regime, its performance degrades in cases where diffusion becomes dominant. This is consistent with the fact that in general, reduction-based AMG methods tend to suffer from growth in complexity and/or convergence rates as the problem size is increased, especially for diffusion dominated problems in two or three dimensions. Motivated by the success of ℓAIR in the advective regime, our aim in this paper is to generalize the AIR framework with the goal of improving the performance of the solver in diffusion dominated regimes. To do so, we propose a novel way to combine mode constraints as used commonly in energy minimization AMG methods with the local approximation of ideal operators used in ℓAIR. The resulting constrained ℓAIR (CℓAIR) algorithm is able to achieve fast scalable convergence on advective and diffusive problems. In addition, it is able to achieve standard low complexity hierarchies in the diffusive regime through aggressive coarsening, something that has been previously difficult for reduction-based methods.

Key words. algebraic multigrid, multigrid reduction, root-node, energy minimization, nonsymmetric, preconditioning

MSC codes. 65N55, 65N22, 65F08, 65F10

1. Introduction.

We design and analyze a reduction-based algebraic multigrid (rAMG) algorithm for linear systems of algebraic equations

$$A\mathbf{u} = \mathbf{f}, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is assumed to be a sparse nonsymmetric matrix. Our focus in this paper is on solving (1.1) arising from discretizations of advection-diffusion-reaction partial differential equations (PDEs), which arise in various practical applications and also serve as interesting initial model problems for testing nonsymmetric AMG solvers. Ultimately, our objective is a method that is naturally robust and efficient in both the advection and diffusion limits.

Multigrid methods for solving (1.1) use a relaxation process or smoother, defined in this paper by M, as a local solver for a sequence of coarse-level systems of equations to reduce the global error resulting from applying relaxation on the finest level. In AMG, a recursive two-level point of view is often used, both in terms of the development of the AMG setup algorithm as well as the analysis of the solver it produces. In this two-level context, the idea is to analyze two complementary processes to efficiently solve sparse linear systems, a relaxation scheme on the fine-level, with corresponding error propagation matrix given by $I - M^{-1}A$, and a coarse-level correction, with error propagation matrix given by $I-PA_c^{-1}RA$. Here, $P \in \mathbb{R}^{n\times n_c}$, $n_c < n$, denotes the interpolation matrix that maps corrections from the coarse-level, $R \in \mathbb{R}^{n_c\times n}$ is the restriction matrix that maps residuals to the coarse-level, and $A_c = RAP$ is the coarse-level operator. The error propagation matrix of the resulting two-level method, from which a multilevel method is defined recursively, reads

$$E_{TG} = (I - M^{-1}A)(I - PA_c^{-1}RA). \tag{1.2}$$

The AMG solver is then defined from this two-level scheme by recursively applying it on the coarse-level to approximate A_c^{-1}, and reduce the errors remaining after applying relaxation on the finer levels.

In AMG, the smoother M is typically fixed to be a simple point-wise method and then R and P are constructed in an automated setup algorithm that takes as input the system matrix A. In this paper,

*Submitted to the editors June 30, 2023.

Funding: This work was funded by NSF grants DMS-2110917 and DMS-2111219.

†Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM, USA (ahsan@unm.edu)

‡Department of Mathematics, Penn State University, University Park, PA, USA (jjb23@psu.edu)

§School of Mathematics and Natural Sciences, Bergische Universität Wuppertal, Wuppertal, Germany (kkahl@uni-wuppertal.de)

¶Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario, Canada (okrzysik@uwaterloo.ca)

‖Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM, USA (jbschroder@unm.edu)

#Los Alamos National Laboratory, Los Alamos, NM, USA (southworth@lanl.gov)

since we consider a reduction-based AMG framework, we assume that the smoother is a point-wise F-relaxation scheme, where the set F denotes the set of fine variables in the coarse-fine (C/F) splitting of the degrees of freedom Ω such that $C \cup F = \Omega$ and $C \cap F = \emptyset$. In this setting, the main tasks in the AMG setup algorithm are to construct the restriction and interpolation matrices R and P such that certain *approximation properties* hold and R and P (and thus $A_c = RAP$) are *sparse* matrices. The latter sparsity requirement implies that the setup procedure can be efficiently applied recursively to $A_c = RAP$ in order to construct an optimal multilevel AMG solver.

In the SPD case, the variational choice $R = P^T$ is used and the weak approximation property for P bounds the convergence rate of the two-level method in the A-norm. Notably, the weak approximation property has as its minimizer the so-called ideal interpolation matrix [12]. This ideal form of interpolation gives rise to the Schur-complement of A as the coarse-level system matrix and, thus, coincides with block Gaussian elimination based on the given coarse-fine splitting. In contrast to classical AMG methods that use a global smoother in the AMG solver (e.g., weighted-Jacobi or lexicographic Gauss-Seidel), reduction based methods that are motivated by this block factorization interpretation choose smoothers that focus on the subspace defined by the fine variables F.

Traditionally, classical AMG methods are very effective for sparse SPD problems, e.g., discretizations of various diffusion and heat conduction problems, and not as effective for nonsymmetric problems, whereas, reduction-based AMG methods have been developed that work well for nonsymmetric problems with a near-triangular (upwinded) structure, e.g., space-time discretizations [26] and advection-dominated PDEs [20, 18]. The parallel-in-time method called multigrid reduction in time (MGRIT) [11] is another recent application of reduction-based multigrid methods, where time integration is recognized as suitable for reduction-based approaches because time is one-dimensional and optimal coarsening is practical; however, MGRIT differs from rAMG methods in using a non-Galerkin coarse grid typically based on rediscretization in time. Interestingly, theory [27, 10, 29] (and practice) has indicated that without special care (e.g., [9]) MGRIT is effective on space-time PDEs with parabolic problems, particularly SPD spatial discretizations, but struggles with hyperbolic PDEs and highly nonsymmetric spatial discretizations.

Indeed, solving more general non-symmetric systems arising from higher dimensional spatial PDEs or space-time PDEs is more complicated in terms of balancing the convergence and complexity of a multilevel solver. A viable approach for solving the latter higher dimensional problems is given by the ℓAIR solver [20]. The approach has been extensively studied and tested for analogous advection-diffusion-reaction model problems we consider here [21, 18, 20], and also demonstrated on more complex advection-dominated physics, e.g., [28, 8]. In general, ℓAIR performs very well for advection-dominated problems, with performance more mixed for diffusive problems, i.e., for problems with strong diffusion ℓAIR suffers from degraded convergence and similar complexity issues as other reduction-based AMG methods for diffusion type problems in higher dimensions. We mention the recent paper [32] which represents the state-of-the-art in research on techniques for improving the performance of reduction-based AMG solvers for diffusion problems. There, the traditional approach of constructing a row-wise approximation of ideal interpolation is considered and the focus is on improving approximations to A_{ff}^{-1} using sparse approximate inverse techniques, similar to the sparse Krylov approximations that have been studied, and very recently used for a highly efficient parallel variation of ℓAIR [8].

In this paper, we combine the reduction-based principles of ℓAIR with the mode constraints of energy-minimization AMG [17, 31, 6, 22, 25, 19]. The ℓAIR algorithm offers sparse, accurate approximations of ideal transfer operators in the strongly advective regime, which also yields excellent AMG approximation properties [21]. In contrast, in diffusive regimes, accurate and sparse approximations of ideal transfer operators are generally not viable due to the density of A_{ff}^{-1}; because no other information is taken into account, the AMG approximation properties of ℓAIR also suffer or the complexity must dramatically increase. Classical and energy-minimization AMG methods suffer from the opposite problem – they tend to offer excellent AMG approximation properties for diffusive problems, but very poor approximation properties in the advective regime [21]. On a high level, this is because the near nullspace of advective operators cannot be represented simply by smoothed constant vectors, the basis for almost all classical AMG and energy minimization methods. Energy minimization methods inadvertently further block their potential by using some form of normal equations to perform energy minimization in the nonsymmetric setting (wherein the nonsymmetric matrix A does not define a natural energy or minimization). Such minimization converges to the ideal transfer operators *of the normal equations* [19], even though a sparse accurate approximation of ideal transfer operators is often viable for matrix A, which ℓAIR successfully

targets.

Here, we recognize this subtlety in approximation properties that guides the success of different AMG methods in different regimes, and define a new constrained variation of the ℓAIR algorithm. Constrained ℓAIR (CℓAIR) directly approximates the ideal transfer operators using a similar objective as in ℓAIR, while constraining the range of P or R^T to include known or expected near-null space mode(s), thereby harnessing the power of reduction-based and energy-minimization methods in their respective regimes. In relaxation, we restrict ourselves to simple F- and C-point Jacobi relaxation, demonstrating that with careful construction of transfer operators, we are able to apply an efficient reduction-based method for diffusive-dominated problems. As it turns out, the construction of sparsity patterns for transfer operators in ℓAIR and root-node based energy minimization [19] are very similar in principle, namely that they are defined column-wise for P and row-wise for R. As a result, we are also able to naturally incorporate an aggressive root-node approach to choosing coarse variables [25, 19] for diffusion dominated problems, ameliorating the (necessarily) high complexity that tends to arise in reduction-based and advective solvers [20].

In the case of anisotropic diffusion, we observe that the energy minimization approach depends crucially on the aggressive root-node coarsening technique [25]. In this case, the aggressive coarsening selects a relatively small number of coarse variables to compensate for the complexity of the long-range interpolation required for the non-grid aligned anisotropy [25]. The outcome is a low-complexity, effective AMG solver. By combining neighborhoods of fine degrees of freedom (DOFs) into a single coarse variable, root-node accomplishes this by greedy aggregation [30], in which the seed point of each aggregate becomes the C-point (root-node) and the remaining degrees of freedom become F-points. The member points (F and C) of each aggregate define the initial nonzero pattern of the corresponding column of P. Generally speaking, this greedy aggregation procedure places C-points farther apart than the traditional Ruge-Stüben (RS) C/F-splitting of the fine degrees of freedom. This results in a more aggressive coarsening and fewer coarse variables (i.e., lower complexity) for root-node [25, 19]. The column-wise viewpoint that we employ in the definition of CℓAIR interpolation allows us to similarly use aggressive root-node coarsening to control complexity, particularly for diffusive problems, and to also incorporate constraint vectors for improved AMG convergence that are fit into span(P) (similar to smoothed aggregation methods [30, 19]).

This paper is organized as follows. The next section reviews reduction-based AMG and the ℓAIR framework that motivates our new method, and provides practical connections to energy minimization AMG methods that facilitate our method design. Our new method CℓAIR is presented in Section 3, and Section 4 contains numerical results that illustrate the performance of our algorithms applied to various discretizations of our model advection-diffusion problem. In particular, CℓAIR is able to maintain fast robust convergence on advection-dominated problems, while also yielding low-complexity scalable solutions for diffusion dominated problems. To complement the numerical results for the proposed method CℓAIR, we present a study of classical AMG weak and strong approximation properties in Appendix A.

2. Reduction based AMG and ℓAIR interpolation. Let $A \in \mathbb{R}^{n\times n}$ and assume that the degrees of freedom $\Omega = \{1, ..., n\}$ are partitioned in the classical sense such that we have n_c C-points and n_f F-points. Then, A can be represented in the following block form:

$$A = \begin{pmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{pmatrix}. \tag{2.1}$$

As before, define $P : \mathbb{R}^{n_c} \mapsto \mathbb{R}^n$ and $R : \mathbb{R}^n \mapsto \mathbb{R}^{n_c}$ as interpolation and restriction respectively. Further, assume that C-points are interpolated and restricted by injection in the classical AMG sense; then, the transfer operators P and R in reduction based AMG can be written in the following block form:

$$P = \begin{pmatrix} W \\ I \end{pmatrix}, \quad R = \begin{pmatrix} Z & I \end{pmatrix}, \tag{2.2}$$

where the ordering here is useful in formalizing a reduction-based AMG method. Design of classical reduction based AMG methods is motivated by the observation that the ideal interpolation operator is the unique operator

$$P_{\text{ideal}} = \begin{pmatrix} -A_{ff}^{-1}A_{fc} \\ I \end{pmatrix} \tag{2.3}$$

that eliminates the contribution of the coarse-grid correction $\mathbf{e}_c$ to the F-point residual:

$$AP\mathbf{e}_c = \begin{pmatrix} \mathbf{0} \\ S\mathbf{e}_c \end{pmatrix}. \tag{2.4}$$

Assuming R and P take the form of (2.2), we have independent of A that the Petrov-Galerkin coarse grid satisfies $RAP = S \coloneqq A_{cc} - A_{cf}A_{ff}^{-1}A_{fc}$, where S is the Schur complement [12, 20].

Reduction based methods for diffusion type problems have traditionally assumed the classical AMG form of interpolation in (2.3) and approximate $-A_{ff}^{-1}A_{fc}$ by solving for each $i \in C$

$$A_{ff}W^{[i]} = -A_{fc}^{[i]}, \tag{2.5}$$

for the n_c columns of the interpolation weight matrix W defined as in (2.2). Here, we use the notation $X^{\{i\}}$ to refer to the ith row of a matrix X, whereas, $X^{[i]}$ will denote a column of the matrix. In this classical reduction-based setting, choosing the coarse grid degrees of freedom C as well as choosing the sparsity structure of the rows of W are done using classical AMG coarsening and strength of connection heuristics and the resulting algorithms typically lead to high grid and operator complexities, even when very simple approximations of A_{ff}^{-1} are used. We mention that various approximations to A_{ff}^{-1} in the computation of P and in relaxation are possible and have been considered in the literature, and we reiterate that in our development we focus on designing an approach that achieves both low grid and operator complexities while at the same time only requiring the simplest diagonal (Jacobi) F-relaxation scheme for fast convergence.

2.1. Review of ℓAIR. The ℓAIR approach that we build our new method around is based on the similar observation that the ideal restriction operator is the unique operator [20]

$$R_{\text{ideal}} = \begin{pmatrix} -A_{cf}A_{ff}^{-1} & I \end{pmatrix} \tag{2.6}$$

that eliminates all error at F-points:

$$RA\begin{pmatrix} \delta\mathbf{e}_f \\ \mathbf{0} \end{pmatrix} = \mathbf{0} \quad \forall \delta\mathbf{e}_f. \tag{2.7}$$

Here, the ordering of the equations is again based on a splitting of $\Omega = C \cup F$. The ℓAIR approach is then based on setting RA equal to zero in (2.7) within a pre-determined F-point sparsity pattern for Z. A similar method to approximate ideal interpolation can be expressed as satisfying $AP = 0$ exactly a specified sparsity pattern for W. Note that the AIR approach can also be seen as directly approximating the action of R_{ideal} on F-points, where $R_{\text{ideal}}A = (\mathbf{0}, S)$, for the Schur complement S. Expressing this result in terms of some matrix Z, the approach is equivalent to satisfying

$$ZA_{ff} = -A_{cf} \tag{2.8}$$

within a predetermined sparsity pattern for Z. Here, the AIR approach is clearly different from the classical (reduction) AMG [4, 24, 16, 13] approach in that (2.8) involves solving for the n_c rows of Z (or R), which now gives a column-wise view of computing R^T and thereby an ℓAIR style form of P. [1]

Denoting indices of the sparsity pattern for the ith row of Z as $\mathcal{Z}_i = \{\ell_1, ..., \ell_{S_i}\}$, where $S_i = |\mathcal{Z}_i|$ is the size of the sparsity pattern, the resulting (transposed) linear system for Z takes the form

$$\begin{pmatrix} a_{\ell_1\ell_1} & a_{\ell_2\ell_1} & ... & a_{\ell_{S_i}\ell_1} \\ a_{\ell_1\ell_2} & a_{\ell_2\ell_2} & ... & a_{\ell_{S_i}\ell_2} \\ \vdots & & \ddots & \vdots \\ a_{\ell_1\ell_{S_i}} & a_{\ell_2\ell_{S_i}} & ... & a_{\ell_{S_i}\ell_{S_i}} \end{pmatrix} \begin{pmatrix} z_{i\ell_1} \\ z_{i\ell_2} \\ \vdots \\ z_{i\ell_{S_i}} \end{pmatrix} = - \begin{pmatrix} a_{i\ell_1} \\ a_{i\ell_2} \\ \vdots \\ a_{i\ell_{S_i}} \end{pmatrix}. \tag{2.9}$$

[1] We note that the reduction-based AMG approaches [16, 13] have also previously explored the use of constraint vector(s), however in addition to our different column-wise view of computing R^T, these approaches use cheap approximations to A_{ff}^{-1} (e.g., diagonal) with a more expensive multilevel adaptive approach for generating the constraint vector. In this work, we take the more expensive ℓAIR approach to approximating A_{ff}^{-1} based on small block inverses, but combine that with a cheap constraint vector similar to energy-minimization methods, which require only a few relaxation sweeps on each level for improvement.

As demonstrated in [20, 18], for upwinded advection-dominated problems, very good sparse approximations to A_{ff}^{-1} can be made, which means that satisfying (2.8) within a sparsity pattern provides an accurate approximation to R_{ideal}. For advection-dominated problems, this approach also provides very good approximation properties of the resulting restriction operator, as demonstrated in [21] and the numerical results. In contrast, relying only on solving (2.8) for a diffusion dominated problem is unlikely to be effective, because A_{ff}^{-1} is generally more dense in this setting and not well approximated by a sparse matrix. Hand-in-hand with this, the resulting approximation properties are also poor. This has been mitigated reasonably well by combining ℓAIR restriction with classical AMG interpolation for problems with strong diffusion, but the complexity remains high and convergence sub-par compared with classical AMG or energy minimization methods.

2.2. Review of Root-node AMG. One way to conceptualize root-node AMG [19] is as a combination of classical and aggregation-based multigrid methods. Root-node AMG employs a hybrid strategy in which smoothed aggregation type strength-of-connection is used and aggregates are created using standard aggregation routines. One node is selected as the 'root-node' in each aggregate which corresponds to a C-point, while all other nodes in the aggregate are identified as F-points. After that, transfer operators are created in the following manner.

Root-node utilizes algebraically smoothed candidates B, which are fit into the span of interpolation, and in order to recurse, coarse versions, B_c, of the candidates are obtained by injecting B to the C-points. Next, the initial tentative interpolation T is formed by injecting only the first q candidates over each aggregate, where q is the block size of the original matrix ($q = 1$ for a scalar problem). On the coarse grid, each root-node thus represents q DOFs. On T, a further step is taken to normalize every column. This procedure produces the following form

$$T = \begin{matrix} W \\ I \end{matrix} \begin{matrix} \} \text{ Non Root-nodes} \\ \} \text{ Root-nodes} \end{matrix} \tag{2.10}$$

For $q = 1$, T has non-overlapping columns; for $q > 1$, W is block diagonal.

The remaining candidates are projected into range(T) in the Euclidean inner-product if there are more than q candidates. It is expected in root-node AMG that the allowed sparsity pattern of T has enough DOFs to make this an underdetermined problem. The sparsity pattern of T is typically grown with strength of connection information, as we later do for CℓAIR. Consequently, a minimal norm update is applied to each row of T, guaranteeing that $TB_c = B$ and T adheres to the sparsity pattern. The interpolation P is then generated using subsequent energy-minimization updates to T (briefly covered in the following subsection). Root-node minimizes energy by solving $A \begin{pmatrix} W \\ I \end{pmatrix} = 0$ subject to the mode interpolation constraints so that the solution is non zero.

2.3. Energy minimization and mode constraints. Another well-known class of AMG methods is that of energy-minimization. We will particularly focus on root-node energy-minimization [19], which shares the C/F-splitting design of reduction-based methods. One key part of energy-minimization is the use of constraints during the minimization process. For efficiency reasons, the sparsity pattern of P is constrained. Let $\mathcal{W}$ be the sparsity pattern for the F-rows in P. We denote that P obeys the sparsity pattern constraint with

$$P \in \begin{bmatrix} \mathcal{W} & I \end{bmatrix} \quad \text{or} \quad W \in \mathcal{W}. \tag{2.11}$$

For approximation property reasons, a near nullspace mode constraint is also typically enforced where

$$B \in \text{span}(P), \tag{2.12}$$

and $B \in \mathbb{R}^{n,k}$ is a set of k global near nullspace modes. With P of the form (2.2) and equation (2.12), this then implies that $B_c = [0\ I]\, B$, i.e., the fine-grid B is injected to the coarse-grid, along with the constraint,

$$PB_c = B. \tag{2.13}$$

Similar to ℓAIR, energy-minimization AMG takes a column-oriented view. Letting $P^{[j]}$ denote column j, construction of transfer operators is based around a minimization along the lines of

$$P = \underset{P}{\text{argmin}} \sum_j \|P^{[j]}\|_{\mathcal{X}}^2, \text{ such that constraints (2.13) and (2.11) are satisfied,} \tag{2.14}$$

where $\mathcal{X}$ denotes some norm, usually A for SPD operators or A^*A for nonsymmetric matrices. This column-oriented view is analogous to ℓAIR in (2.9), but here we are minimizing in some energy-induced inner product, rather than solving each block equation exactly as in ℓAIR.

Interestingly, this distinction has more profound consequences for the efficacy of the methods. For SPD operators, root-node uses projected conjugate gradients (CG) for equation (2.14); without constraints (2.13) and (2.11), such a minimization procedure converges to P_{ideal} for A [19, Lemma 4.2]. However in the nonsymmetric case, root-node uses a projected generalized minimal residual (GMRES) for equation (2.14), which in turn (without constraints (2.13) and (2.11)), converges to P_{ideal} for A^*A [19, Lemma 4.6]. Indeed, by posing the unconstrained problem as an (overdetermined) minimization, root-node is required to define an energy-induced inner product through the normal equations, which in turn leads to the approximation of P_{ideal} for the normal equations rather than directly for the operator of interest. In contrast, the base algorithm of ℓAIR directly approximates (and converges to) P_{ideal} for the original operator A. This is possible because the algorithm is built around local matrix approximation (2.8) rather than a matrix-induced norm, which does not naturally exist for non-SPD matrices.

As it turns out, which ideal operators we are trying to approximate (without constraints) is an important distinction for highly advective problems, which typically generate discretization matrices that are close to block lower-triangular in some ordering [18]. For such cases [18], ℓAIR achieves good sparse approximations to A_{ff}^{-1} for computing R_{ideal} and overall excellent AMG convergence. However if one were to approximate ideal transfer operators based on A^*A instead of A, the block lower-triangular structure that is key to achieving sparse and accurate approximations to A_{ff}^{-1} and R_{ideal} is completely lost. The natural result is that a good sparse approximation to A_{ff}^{-1} becomes more difficult to compute (see [18], Section 4 for more discussion), and the resulting AMG method is significantly less effective. Thus although the underlying problem that energy-minimization is based around, namely approximating $AP = \mathbf{0}$, is almost equivalent to ℓAIR, by formulating via energy minimization the resulting class of methods yield lackluster performance on highly advective problems.

2.4. The best of both worlds. Looking carefully at the ℓAIR and energy-minimization approaches leads us to consider a new interpretation combining the best of both worlds. By directly approximating R_{ideal} of the original operator A, ℓAIR is able to construct highly effective transfer operators for advection-dominated problems; in contrast, mode constraints are fundamental to the efficacy of energy-minimization methods for diffusion dominated problems (indeed, without constraints energy minimization alone is generally not effective). Although ℓAIR is based around approximation of ideal transfer operators, we can also think in terms of approximation properties – consider each row of R as the local fine-grid mode being restricted to a given C-point, where these modes should be local representations of the smooth error. This is exactly what happens in classical smoothed aggregation [30], as well as when bilinear interpolation is used in geometric MG for diffusion. Thus we propose a new constrained ℓAIR (CℓAIR) method that is built around directly approximating the ideal transfer operators of A in an ℓAIR framework, regardless of whether A is SPD or nonsymmetric, while also incorporating mode constraints as in energy minimization to improve robustness in diffusion dominated problems.

In summary, the proposed CℓAIR approach combines strengths of root-node and ℓAIR, with the goal of a robust solver in both the advective and diffusive regimes. The CℓAIR approach for constructing ℓAIR interpolation with constraints is directly related to solving (2.8) and thus is a constrained approximation of P_{ideal} for the original operator, in contrast to root-node, which targets the normal equations in the nonsymmetric setting. Also, all of our proposed generalizations can be used to build R and/or P.

3. Constrained ℓAIR transfer operators. The main goals of our new reduction-based AMG method built around ℓAIR-style interpolation, is to have a solver that (i) works well for both advection and diffusion problems, (ii) allows for the incorporation of mode interpolation constraints (local or global), and (iii) controls complexity in a reduction setting for diffusive problems through an aggressive root-node coarsening. To achieve these goals, we consider mixing ideas from ℓAIR, which works well for advection, with energy-minimization and smoothed aggregation (SA) [30], which work well for anisotropic diffusion problems and allow for mode constraints. Another key component of the algorithm for diffusion dominated problems, described in detail in a latter section, is our use of a root-node aggregation-based coarsening algorithm.

The overall ℓAIR interpolation scheme we consider is described as follows. Similar to the energy-minimization discussion above, we will enforce that P obeys the sparsity pattern constraint (2.11). Regarding the sparsity pattern $\mathcal{W}$ for the F-rows, let $\mathcal{W}_i$ denote the sparsity pattern for the ith column

of W, that is $\mathcal{W}_i = \{m_1, \ldots, m_{T_i}\}$ and the number of nonzeros in column i equals $T_i = |\mathcal{W}_i|$. Define $A_{ff}^{(i)}$ as A_{ff} restricted (in rows and columns) to the sparsity pattern $\mathcal{W}_i$, and $a_{fc}^{[i]}$ and $w^{[i]}$ as A_{fc} and W restricted to column i, respectively, with rows restricted to the sparsity pattern of $\mathcal{W}_i$. The standard ℓAIR approach for finding each $w^{[i]}$ is then equivalent to solving: for each $i \in n_c$ solve

$$A_{ff}^{(i)} w^{[i]} = -a_{fc}^{[i]}. \tag{3.1}$$

If we consider solving (3.1) at each i, we can rewrite the procedure as the following global block diagonal system, where we assume n_c points on the coarse grid:

$$A_{ff}^{(*)} w^{[*]} := \begin{bmatrix} A_{ff}^{(0)} & & & \\ & A_{ff}^{(1)} & & \\ & & \ddots & \\ & & & A_{ff}^{(n_c)} \end{bmatrix} \begin{bmatrix} w^{[0]} \\ w^{[1]} \\ \vdots \\ w^{[n_c]} \end{bmatrix} = - \begin{bmatrix} a_{fc}^{[0]} \\ a_{fc}^{[1]} \\ \vdots \\ a_{fc}^{[n_c]} \end{bmatrix} = -a_{fc}^{[*]}. \tag{3.2}$$

The solution of system (3.2) gives us classic AIR interpolation. As a side note, let W be stored in a sparse matrix format, then the global vector of all the nonzeros of W, denoted $w^{[*]}$, corresponds to the data array for the sparse representation of W when stored in compressed column format.

3.1. Proposed Method with Global Constraints. Given the above formulation of ℓAIR in (3.2), we define the procedure for incorporating a global mode interpolation constraint into the approach. To enforce the mode interpolation constraint (2.13), define the matrix Q, such that $Qw^{[*]}$ is equivalent to PB_c, i.e., the entries of B_c populate Q such that the constraint equation (2.13) is equivalent to saying

$$Qw^{[*]} = \begin{bmatrix} B^{[0]}|_F \\ B^{[1]}|_F \\ \vdots \\ B^{[k]}|_F \end{bmatrix} = B^{[*]}|_F, \tag{3.3}$$

where $B^{[i]}$ is the ith column of B, $B^{[*]}$ represents the columns of B stacked vertically, and $B^{[*]}|_F$ represents columns of B stacked vertically, but restricted to the fine-grid points F. Equation (3.3) is also equivalent to the constraint that $WB_c = B|_F$.

Thus our constrained minimization problem is

$$\min_{w^{[*]}} \left\| a_{fc}^{[*]} + A_{ff}^{(*)} w^{[*]} \right\|_2 \tag{3.4a}$$

$$\text{subject to } Qw^{[*]} = B^{[*]}|_F. \tag{3.4b}$$

This is an equality constrained minimization problem, with various solution approaches [14].

3.1.1. Direct Solution to Minimization Problem. The minimization problem (3.4a)–(3.4b) can be solved directly via the following Karush–Kuhn–Tucker (KKT) system

$$\begin{bmatrix} (A_{ff}^{(*)})^T A_{ff}^{(*)} & Q^T \\ Q & \mathbf{0} \end{bmatrix} \begin{bmatrix} w^{[*]} \\ \lambda \end{bmatrix} = \begin{bmatrix} -(A_{ff}^{(*)})^T a_{fc}^{[*]} \\ B^{[*]}|_F \end{bmatrix}. \tag{3.5}$$

The (1,1) block of equation (3.5) uses the normal equations, as the minimization principle requires an SPD matrix and we do not assume that $A_{ff}^{(*)}$ is SPD.

System (3.5) could be solved exactly via a Schur complement approach. If this is done, letting $\bar{A}^{-1} = \left((A_{ff}^{(*)})^T A_{ff}^{(*)} \right)^{-1}$, the solution is

$$w^{[*]} = \left(I - \bar{A}^{-1} Q^T (Q \bar{A}^{-1} Q^T)^{-1} Q \right) \bar{A}^{-1} (A_{ff}^{(*)})^T a_{fc}^{[*]} \; + \; \left(\bar{A}^{-1} Q^T (Q \bar{A}^{-1} Q^T)^{-1} Q^T \right)^{-1} B^{[*]}|_F, \tag{3.6}$$

where Q is the rectangular constraint matrix from above. Upon inspection, computing (3.6) is an expensive endeavour, especially $(Q\bar{A}^{-1}Q^T)^{-1}$ and potentially, the transpose of $A_{ff}^{(*)}$. To avoid these costs, we consider an iterative approach related to energy-minimization AMG [22].

3.1.2. Iterative Solution to Minimization problem. A review of approaches for obtaining an inexpensive iterative solution to (3.5) is given in [14]. One option previously used for AMG (e.g., for root-node) is projected Krylov methods. Here, an initial guess (tentative interpolation) $w^{[*],t}$ that satisfies the constraints is constructed, so that $Qw^{[*],t} = B^{[*]}|_F$. Then, a projected Krylov method using Q is applied to solve the interpolation equation (3.2). The inverse $(Q\bar{A}^{-1}Q^T)^{-1}$ is not required and the transpose is not needed, because we only compute the residual for the interpolation equation (3.2) when computing a descent direction for equation (3.4a). The previous works [22, 19] use such a projected CG and GMRES approach for the symmetric and nonsymmetric cases, respectively. However as noted in Section 2.3, the nonsymmetric GMRES approach will approximate P_{ideal} in the constraint space for the normal equations, which is not desirable.

Thus here, we consider a simpler and cheaper linear iteration for minimizing (3.4a) that approximates P_{ideal} for the original operator in the constraint space. We find that this approach yields effective restriction and interpolation operators.

An additional cost consideration is whether or not to precondition such an iterative solve. Since the matrix $A_{ff}^{(*)}$ is block diagonal, each block could be inverted (or approximately inverted). Thus, we consider the use of approximate inverse preconditioners of the following form:

$$A_{ff}^{(*),-1} \approx \widehat{A_{ff}^{(*),-1}} = \begin{bmatrix} \widehat{A_{ff}^{(0),-1}} & & & \\ & \widehat{A_{ff}^{(1),-1}} & & \\ & & \ddots & \\ & & & \widehat{A_{ff}^{(n_c),-1}} \end{bmatrix}, \tag{3.7}$$

where $\widehat{A_{ff}^{(i),-1}}$ represents an approximate inverse to that block. Importantly, this inverse is *local* and can be generated in a variety of ways, e.g., diagonal, GMRES, or ILU approximations to each individual block inverse. This is *in contrast* to the "classic" energy-minimization which uses a single global Krylov polynomial to simultaneously approximate all block inverses. It is our belief that the block inverse approach is more effective. In particular, a global Krylov polynomial effectively assumes that each block has the same minimizing polynomial, whereas in reality each block will likely have its own unique minimizing polynomial (distinct from other blocks). The local approach allows us to calculate more accurate local inverses faster through locally accurate approximations and polynomials, or to solve each local equation directly.

3.1.3. Proposed Algorithm for Computing R and P. We now present our simple iterative scheme for minimizing (3.4a)–(3.4b) in Algorithm 3.1. The approach is a projected one-step iteration, which iteratively finds AIR-like interpolation operators with constraints. For restriction, the simplest approach on paper applies Algorithm 3.1 to A^T. However if forming a transpose is computationally expensive, one can also reformulate Algorithm 3.1 relative to $RA = 0$, as in the original ℓAIR method [20]. Then, the algorithm will still extract small submatrices $A_{ff}^{(i)}$, but after the extraction these submatrices will be transposed.

As input, the algorithm takes the operator A and corresponding strength of connection matrix S. Unless noted otherwise, we use the classical strength measure [24]. The input sparsity degree pattern m determines how wide the sparsity pattern in P will be, with $m = 1$ corresponding to distance-one interpolation based on the sparsity pattern of S_{fc}. Most commonly, we will use $m = 2$, which expands the sparsity to consider degree-two connections, similarly to ℓAIR and root-node. Next, the input "Coarsen type" considers whether a classical "FC" coarsening, e.g., Ruge-Stüben coarsening [24], or an aggregation-based coarsening is used.

The coarsen type controls the base sparsity pattern for P. If classical FC coarsening is used, then the base pattern T comes from the FC rows of the strength matrix, S_{fc}. If an aggregation-based coarsening is used, which is significantly more aggressive, then the "Aggregation Operator"[2] is used for T. This

[2] The "Aggregation Operator" is generated by the algorithm from [30]. First, an aggregation (disjoint splitting) is computed with a greedy graph algorithm that finds the next degree-of-freedom (root node) with all unmarked neighbors and places those degrees-of-freedom in an aggregate and then marks them. A clean-up phase takes all unmarked degrees-of-freedom and places them in an adjacent aggregate. The root node of each aggregate is treated as a C-point. This procedure produces a C/F-splitting that is significantly more aggressive (fewer C-points) than is typical for classical AMG

base pattern T is then expanded based on the strength of connection matrix S via $m-1$ multiplications in line 13. We note that basing the interpolation pattern on strong connections is the same strategy as used by ℓAIR and root-node, and this approach allows us to generate nearly identical patterns.

The least squares solution for line 15 is computed using a psuedoinverse based on B_c restricted to the sparsity pattern of row i, $w^{\{i\},t}$. These pseudoinverses can be locally precomputed for efficiency and are typically small.

The projection operation in line 19 takes each new update $\bar{w}^{[*]}$ and projects it so that $Q\bar{w}^{[*]} = 0$. That is, this operation ensures that each update $\bar{w}^{[*]}$ does not disturb the mode interpolation relationship $Qw^{[*],t} = B^{[*]}|_F$. The projection operation with Q can be implemented locally using the same strategy as for line 15 (see [22]).

Algorithm 3.1 CℓAIR Algorithm

1: **Input**: A: Matrix
2: S: Strength matrix for interpolation
3: B: User supplied mode constraint vector(s)
4: m: Sparsity pattern degree
5: FC or Agg: Coarsening type
6: **Output**: P: Interpolation in the form of the weight block $w^{[*],t}$
7: **set** tentative prolongation $w^{[*],t}$, corresponding to $[-A_{fc}, I]$
8: **set** base sparsity pattern T based on coarsening type
9: **if** Coarsen type is FC
10: $T \leftarrow [S_{fc}, I]$
11: **else if** Coarsen type is Agg
12: $T \leftarrow$ Aggregation Operator
13: **set** expanded sparsity pattern to match F-row structure of $S^{m-1}T$
$\mathcal{W} \leftarrow$ sparsity_pattern $\left((S^{m-1}T)|_F\right)$
14: **expand** $w^{[*],t}$ to store (possibly zero) entries for every nonzero in $\mathcal{W}$
15: **enforce** constraints on $w^{[*],t}$ such that $Qw^{[*],t} = B^{[*]}|_F$, by taking row i, $w^{\{i\},t}$, and computing
$w^{\{i\},t} \leftarrow$ least squares solution to $w^{\{i\},t}B_c = B$
16: **compute** exact or approximate block inverses for $\widehat{A_{ff}^{(*),-1}}$
17: **for** $k = 1, 2, \ldots$ **do**
18: **compute residual update** to minimize equation (3.4a), using $\mathbf{0}$ initial guess for $\bar{w}^{[*]}$:

$$\bar{w}^{[*]} \leftarrow \widehat{A_{ff}^{(*),-1}} \left(\left(-a_{fc}^{[*]} - A_{ff}^{(*)} w^{[*],t} \right) - A_{ff}^{(*)} \mathbf{0} \right)$$

19: **project update**: $\bar{w}^{[*]} \leftarrow (I - Q^T(QQ^T)^{-1}Q)\bar{w}^{[*]}$
20: **update interp**: $w^{[*],t} \leftarrow w^{[*],t} + \bar{w}^{[*]}$
21: **end for**
22: **set** final interpolation weights: $w^{[*]} \leftarrow w^{[*],t}$
23: **Return** $w^{[*]}$

Remark 3.1. If the constraints are removed from Algorithm 3.1 and the exact inverse $A_{ff}^{(*),-1}$ is used, then ℓAIR is recovered. That is, removing the constraint lines 15 and 19 and assuming $k = 1$ yields the final update in line 20 of

$$\begin{aligned} w^{[*],t} &\leftarrow w^{[*],t} + w^{[*],1} \\ &\leftarrow w^{[*],t} + A_{ff}^{(*),-1} \left(-a_{fc}^{[*]} - A_{ff}^{(*)} w^{[*],t} \right) \\ &\leftarrow -A_{ff}^{(*),-1} a_{fc}^{[*]} \end{aligned}$$

or rAMG. The "Aggregation Operator" is then a binary matrix where column i corresponds to aggregate i (i.e., the ith C-point) and this column is nonzero only for the degrees-of-freedom in aggregate i. See also [19] which constructs initial sparsity patterns in this manner.

However, we always use the constraints in Algorithm 3.1, making CℓAIR distinct from ℓAIR.

Remark 3.2. We will most often use the exact inverse $A_{ff}^{(*),-1}$ for $\widehat{A_{ff}^{(*),-1}}$ in line 16, similar to ℓAIR. In this case, Algorithm 3.1 is run with $k = 1$, i.e., the output of the algorithm does not change for $k > 1$.

3.1.4. Comparison to Root-node. We now clearly distinguish the similarities and differences between CℓAIR and root-node. The CℓAIR approach shares with root-node (i) the same mode and sparsity constraints (2.13) and (2.11), (ii) the ability to iteratively find P, and (iii) approximates some form of P_{ideal}.

Regarding differences, root-node uses a simple diagonal approximation to A_{ff}^{-1}, whereas CℓAIR uses a potentially much more powerful approximation $\widehat{A_{ff}^{(*),-1}}$, where each block is often inverted exactly[3]. The proposed CℓAIR method also uses a simple one-step iteration that is cheaper computationally than root-node, with no required global storage of Krylov vectors or the additional computations and communications required to maintain the (globally) orthogonal Krylov basis. Lastly, root-node has only considered fast aggregation-based coarsening, whereas CℓAIR will support and explore both fast and slow coarsening, targeting diffusive and advective problems, respectively.

4. Numerical Results. We now present supporting numerical results for CℓAIR, comparing against root-node and ℓAIR for a variety of classic model diffusion problems and for advection-diffusion problems over a range of diffusion parameters. The proposed solver is shown to (i) achieve more desirable operator complexities than ℓAIR, (ii) compare well against the root-node solver for symmetric diffusion problems, where root-node is already known to perform well, and (iii) be more robust regarding parameter tuning when compared to ℓAIR, e.g., for the advection-diffusion problems when going from the purely advective to highly diffusive regimes.

Our basic test framework is as follows. The ℓAIR and root-node solvers use the library implementations in PyAMG [3] and CℓAIR is also implemented in PyAMG[4]. We use V(1,1) cycles as a preconditioner for CG and GMRES in the symmetric and nonsymmetric settings, respectively. The smoother choices respect the reduction framework. For presmoothing, we use 1 iteration of CFF-weighted-Jacobi, i.e., one Jacobi sweep on the C-points, followed by two Jacobi sweeps on the F-points. The weight is equal to $1/\rho(D^{-1}A)$. (Here, $\rho(\cdot)$ denotes the spectral radius, which is approximated numerically by 15 iterations of Arnoldi.) For postsmoothing, we use 1 iteration of FFC-weighted-Jacobi, which is defined analogously to CFF. Such relaxation methods are chosen for simplicity, parallelism, and the preservation of a symmetric preconditioner when A is symmetric, so that CG can be used. The absolute halting criteria is 10^{-9} for the smallest problem size, and is then scaled to simulate a discrete L2-norm[5]. We report preconditioned Krylov iterations and work-per-digit of accuracy. One work unit is defined to be the cost of a fine-grid matrix-vector product, and work-per-digit of accuracy estimates how many work units are required to reduce the residual by one order of magnitude. Work measurements allow for easy comparison across methods. To derive our work-per-digit measure, we first estimate the total cost of doing one matrix-vector product at each level in the hierarchy, relative to one matrix-vector product on the finest level. This yields the operator complexity measure

$$\text{OC} = \sum_k \text{nnz}(A_k)/\text{nnz}(A_0), \tag{4.1}$$

where A_k is the operator on level k in the multigrid hierarchy and nnz$(\cdot)$ stands for the number of nonzero entries. To account for the cost of doing relaxation at each level, we multiply OC by 3.5 because we estimate (roughly) the cost of CFF- and FFC-Jacobi to be slightly less than 4 matrix-vector products. Thus work-per-digit of accuracy is estimated as

$$\text{wpd} = 3.5\ \text{OC}/\log_{10}(\gamma), \tag{4.2}$$

where the average residual convergence factor is $\gamma = (\|r_k\|/\|r_0\|)^{1/k}$, r_k is the final residual, and r_0 is the initial residual.

---

[3]Note that CℓAIR also supports using a diagonal inverse similar to root-node, where each block of $\widehat{A_{ff}^{(*),-1}}$ would be the local diagonal inverse. However, this approximation does not always lead to effective AMG hierarchies in our experiments, e.g., for the 3D Poisson problem. Developing more effective approximations is future research.

[4]The CℓAIR implementation is in the `CF_rootnode` branch, commit bf56195b55a27bd99625c385ee9ba1df8814f764.

[5]In 2D, the tolerance is scaled by 2 each grid refinement, and in 3D, by $\sqrt{8}$.

4.1. Diffusion Tests. For the symmetric case, we consider a variety of classic diffusion tests.

2D Poisson. The PDE here is $-u_{xx} - u_{yy} = f$, with Dirichlet boundary conditions on the unit box. The discretization is classic second-order 5-point finite differencing on a regular grid.

3D Poisson. The PDE here is $-u_{xx} - u_{yy} - u_{zz} = f$, with Dirichlet boundary conditions on the unit box. The discretization is classic second-order 7-point finite differencing on a regular grid.

2D Grid-Aligned Anisotropic Diffusion. The PDE here is

$$-\nabla \cdot Q^T D Q \nabla u = f \quad \text{for } \Omega = [0,1]^2, \tag{4.3a}$$

$$u = 0 \quad \text{on } \partial\Omega, \tag{4.3b}$$

where Ω is the unit box domain and

$$Q = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} \tag{4.4}$$

represent the rotation by angle φ and the strength of anisotropy ϵ. The discretization uses a regular grid and bilinear (Q1) finite elements. For this problem $\varphi = 0$ and $\epsilon = 0.001$, representing strong grid-aligned anisotropy in the x-direction.

2D Rotated Anisotropic Diffusion. Here the PDE and discretization are the same as for equations (4.3a)–(4.3b), but $\varphi = \pi/8$ and ϵ remains 0.001. This represents strong non-grid-aligned anisotropy at the angle of $\pi/8$. As noted in [25, 15, 5], this is a difficult discretization and angle of anisotropy for AMG.

Box-in-Box Coefficient Jump. The PDE here is

$$-\nabla \cdot d(x,y) \nabla u = f \quad \text{for } \Omega, \tag{4.5a}$$

$$u = 0 \quad \text{on } \partial\Omega, \tag{4.5b}$$

where $d(x,y)$ is the jumping coefficient. Here, $\Omega = [0,1]^2$, $d(x,y) = 1$ if $(x,y) \notin [0.44, 0.52]^2$, and $d(x,y) = 10^4$ if $(x,y) \in [0.44, 0.52]^2$. The grid is regular and the coefficient jumps are grid-aligned on the finest level, but will not be grid-aligned at coarser levels due to the algebraic coarsening. The discretization is classic second-order 5-point finite differencing from [1] for coefficient jump problems.

Sawtooth Coefficient Jump. Here, the PDE and discretization are the same as for equations (4.5a)–(4.5b), but $\Omega = [0,16]^2$, $d(x,y) = 1$ for points outside the shaded region of Figure 1, and $d(x,y) = 10^4$ for points inside the shaded region [1].

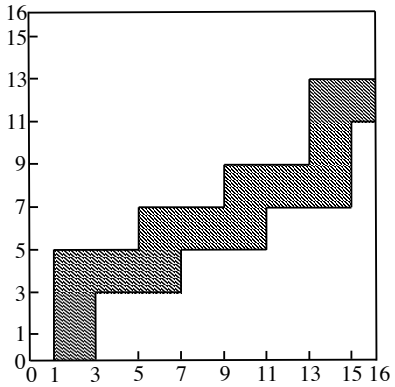
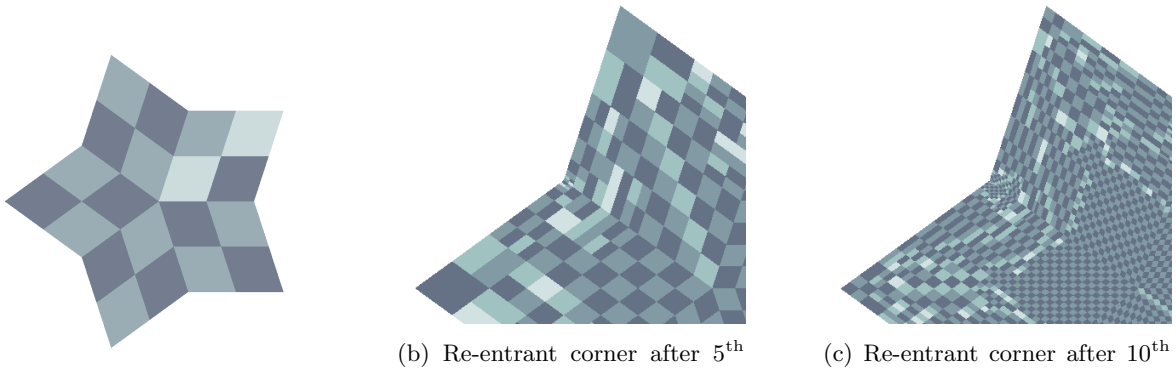


Fig. 1: Sawtooth coefficient jump domain, $d(x,y) = 10^4$ in shaded region and $d(x,y) = 1$ outside.

Laplace Problem with Adaptive Mesh Refinement (AMR). Here, we consider a finite element discretization of the Laplace problem $-\Delta u = 1$ with homogeneous Dirichlet boundary conditions. This problem is solved on a sequence of meshes in an $\mathbb{H}^1$-conforming finite element space which are locally refined in accordance with a simple Zienkiewicz-Zhu [33] error estimator. We save the stiffness matrix A throughout each iteration of the AMR loop and use this matrix to further illustrate differences in outcomes across various AMG solvers. This example is identical to PyMFEM library example 6 [2, 23]. AMR stages for the star mesh file used in this problem are shown in Figure 2.



(a) Initial star mesh

(b) Re-entrant corner after 5^{th} refinement

(c) Re-entrant corner after 10^{th} refinement

Fig. 2: AMR throughout different stages for the Laplace problem.

Standard Solver Parameters. We now list our standard solver parameters for all three methods, ℓAIR, CℓAIR, and root-node. We will occasionally tune the parameters (e.g., strength) for ℓAIR and root-node, in order to make the existing methods more competitive. The parameters for the proposed CℓAIR method will remain fixed for all symmetric test cases, except for the 2D Rotated Anisotropic Diffusion case, where we will follow the guidance of [25] and consider larger degree interpolation sparsity patterns. We use accelerated CG for CℓAIR and root-node and accelerated GMRES for ℓAIR, as the typical approach for ℓAIR is not symmetric (i.e., does not satisfy $R = P^T$, see below parameter choices).

Parameters for ℓAIR (see [20] for more details on parameter definitions)
- Strength-of-connection tolerance for coarsening 0.25
- Degree 2 ℓAIR restriction (sparsity pattern similar to $m = 2$ in Algorithm 3.1) with interpolation strength tolerance 0.05
- Ruge-Stüben coarsening first-pass only [24] (coarsen type FC in Algorithm 3.1, relatively slow coarsening often used by ℓAIR)
- Coarse-grid matrices filtered with drop-tolerance of 10^{-4}
- Classical interpolation formula used for P [24], which is typical for ℓAIR and diffusive problems

Parameters for CℓAIR
- Strength-of-connection tolerance for coarsening 0.5
- Degree 2 sparsity pattern for P ($m = 2$ in Algorithm 3.1) with interpolation strength tolerance 0.5 used to generate S in Algorithm 3.1
- Mode constraint vector $B = \mathbf{1}$, presmoothed with 5 iterations of CFF-weighted-Jacobi at each level
- $R = P^T$ used (see discussion below)
- Aggregation based coarsening (coarsen type Agg in Algorithm 3.1)

Parameters for root-node (see [19] for more parameter details)
- Strength-of-connection tolerance for coarsening 0.25
- P smoothed with energy-minimizing projected CG and a degree 2 sparsity pattern (sparsity pattern similar to $m = 2$ in Algorithm 3.1)
- Mode constraint vector $B = \mathbf{1}$, presmoothed with 5 iterations of Jacobi at each level
- $R = P^T$ used (see discussion below)

– Aggregation based coarsening (standard coarsening for root-node)

We note that the interpolation strength-of-connection tolerance for CℓAIR is different than that for ℓAIR (0.5 versus 0.05). Both of these tolerance were tuned individually for each method. We also note that CℓAIR and root-node both use $R = P^T$, while ℓAIR uses classical interpolation. The use of classical interpolation (instead of R^T, the transpose of ℓAIR restriction) with ℓAIR is typical for diffusive problems, and additionally, the use of R^T as interpolation leads to undesirably large operator complexities.

4.1.1. Poisson Results. We next examine convergence results for the 2D and 3D Poisson problems in Figure 3. Here, we tune ℓAIR to obtain a more challenging baseline solver and set the sparsity pattern in R to degree 1. For these simplest of problems, this choice reduces the operator complexity, while not affecting convergence. For our other test problems, such a parameter choice does not lead to the best performance. Operator complexities over all test problems, which are an advantage of CℓAIR, are discussed later in Section 4.1.3.

Figures 3a and 3c depict results for the 2D and 3D Poisson problems, respectively. Overall, we see flat iteration counts for all methods, except a slight growth for ℓAIR in the 2D Poisson case. The chief performance difference is that ℓAIR has a significantly higher operator complexity than CℓAIR and root-node, which will be discussed in Section 4.1.3

Figure 3b demonstrates that CℓAIR can also be run similar to ℓAIR with Ruge-Stüben coarsening with first-pass only and classical interpolation for P. In general, this setup does not lead to the most efficient solver for CℓAIR, due to the high operator complexity (OC ≈ 3.3).

4.1.2. Other Diffusion Results. We now consider our other diffusion test problems in Figure 4, which again depicts iterations and work-per-digit of accuracy. For these tests, we tune root-node and ℓAIR on the 2D Grid-Aligned Anisotropic Diffusion problem, in order to obtain more challenging baseline solvers. Here, we set the coarsening and interpolation strength-of-connection tolerances to 0.5 and note that using such a high interpolation strength tolerance typically hurts ℓAIR performance, with this case being the outlier. Thus, this is not a general parameter setting for ℓAIR and is used to highlight the greater flexibility of the untuned CℓAIR solver. Following the work [25], we also consider higher-degree interpolation for the 2D Rotated Anisotropic Diffusion case, for both root-node and CℓAIR. Unfortunately, prohibitive increases in operator complexity for ℓAIR do not allow for such an examination of higher-degree interpolation.

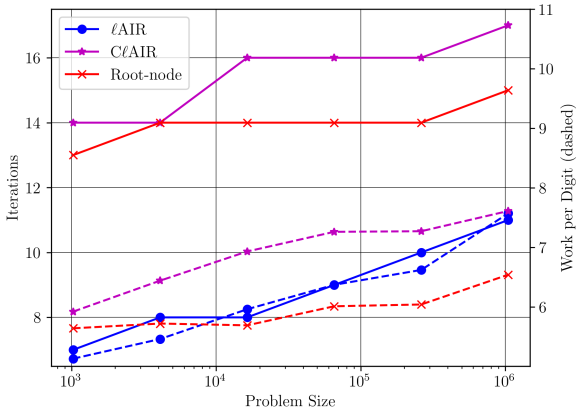
Figure 4a depicts results for the 2D Grid-Aligned Anisotropic Diffusion problem, where we see similar performance for CℓAIR, root-node, and ℓAIR. Figure 4b depicts results for the 2D Rotated Anisotropic Diffusion problem where all three solvers produce similar work-per-digit numbers, and the root-node and CℓAIR solvers show flat, scalable iteration counts for the higher-degree interpolation option.

Figures 4c and 4d depict results for the Box-in-Box Coefficient Jump and Sawtooth Coefficient Jump problems. All three solvers again show similar work-per-digit numbers, with the main difference being in operator complexity, which we discuss in the next subsection.

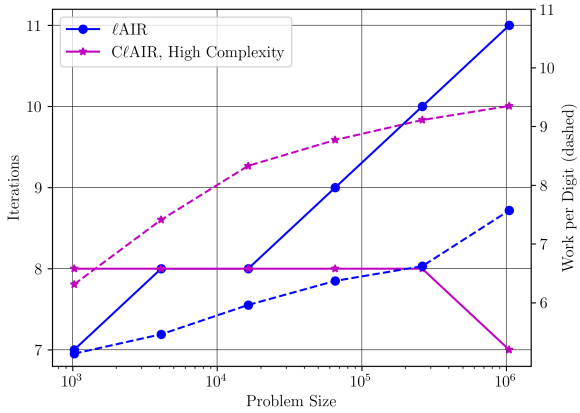
Finally, results for the Laplace problem with AMR utilizing an unstructured star mesh are shown in Figure 4e. While CℓAIR and root-node performance are comparable in this instance, ℓAIR shows high preconditioned iterations and work-per-digit accuracy as problem size increases. ℓAIR performance may be improved by raising the strength-of-connection tolerance from 0.25 to 0.5 and adding the second pass of Ruge-Stüben coarsening, resulting in a maximum of 25 preconditioned iterations. However, in doing so, the operator complexity becomes prohibitively high at around 3.88. Lastly, we note that the work per digit for CℓAIR and root-node grows slowly for this problem as the mesh is adaptively refined, but we do not find this surprising given the re-entrant corners and adaptive refinement. We also observed a similar slow growth when using the benchmark classical Ruge-Stüben solver for this problem.

4.1.3. Operator Complexity Comparison. One key advantage for CℓAIR when compared to other reduction-based approaches on diffusion problems is the ability to achieve moderate operator complexities with good convergence. To illustrate this, Table 1 depicts the operator complexity for all three solvers and the largest problem size for each of the diffusion test problems. Here, this advantage to CℓAIR becomes obvious. The faster aggregation-based coarsening, which only root-node and CℓAIR support, allows for dramatically lower operator complexities and the associated lower storage requirements.

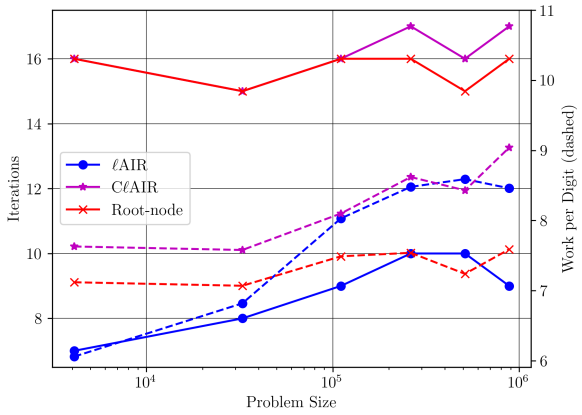
Importantly, one could not simply introduce the aggregation-based coarsening into ℓAIR and maintain good convergence. The addition of the mode interpolation constraint vector B is needed for good



(a) Poisson 2D



(b) Poisson 2D, showing high complexity CℓAIR



(c) Poisson 3D

Fig. 3: 2D and 3D Poisson, comparison of iterations and work-per-digit of accuracy for ℓAIR, CℓAIR, and root-node.

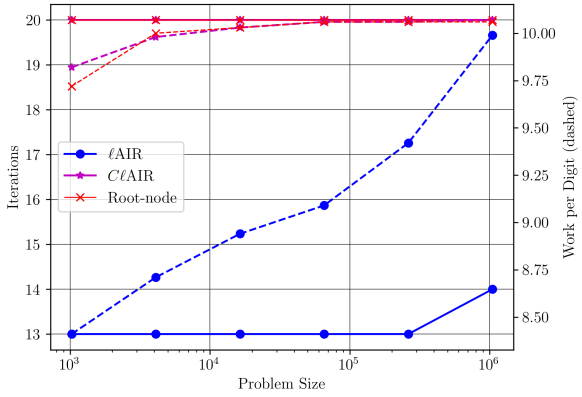
convergence.

We note that the operator complexity of 1.98 for CℓAIR and the 2D Rotated Anisotropic Diffusion case can easily be brought down to 1.78 without a meaningful effect on convergence by using 0.25 as the strength-of-connection tolerance, but we choose to maintain uniform parameters for CℓAIR.

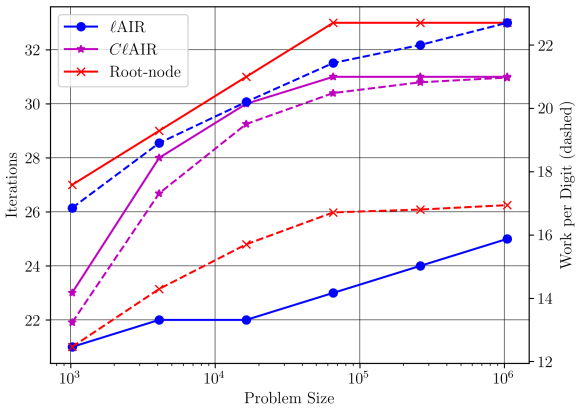
Test Problem	ℓAIR OC	CℓAIR OC	Root-node OC
2D Poisson	2.20	1.40	1.34
3D Poisson	2.87	1.71	1.57
2D Grid-aligned Anisotropic Diffusion	4.92	1.52	1.51
2D Rotated Anisotropic Diffusion	2.89	1.98	1.52
Box-in-Box Coefficient Jump	2.73	1.40	1.34
Sawtooth Coefficient Jump	2.71	1.42	1.35
Laplace Problem with AMR	2.03	1.32	1.17

Table 1: Operator complexities of ℓAIR and CℓAIR for different diffusion test problems.

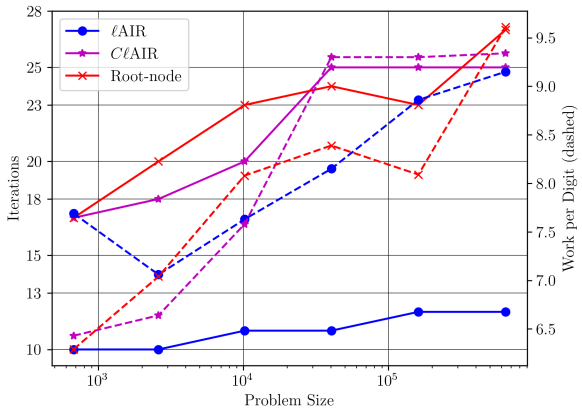
Remark 4.1. In summary, we have shown that CℓAIR performs similarly in terms of work-per-digit, operator complexity, and iterations on this suite of classic diffusion test problems when compared to



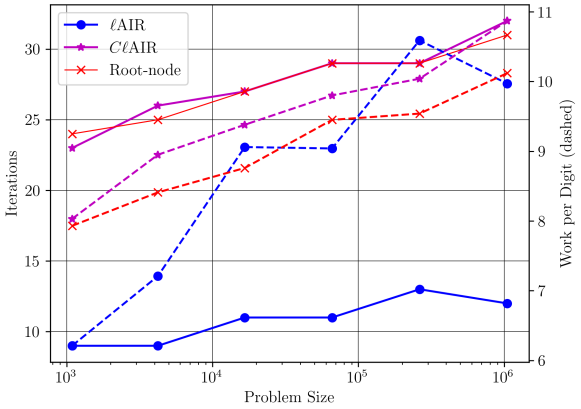
(a) 2D Grid-aligned Anisotropic Diffusion, when not visible, the lines for CℓAIR are underneath root-node.



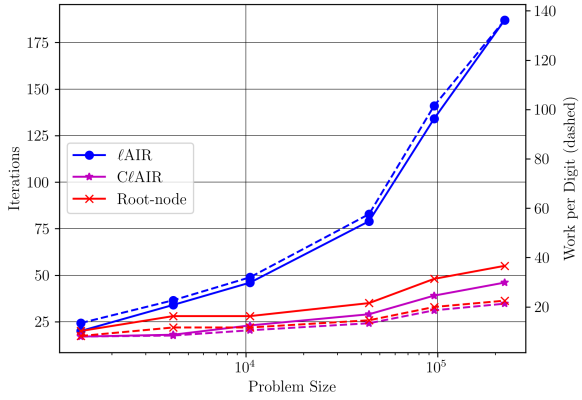
(b) 2D Rotated Anisotropic Diffusion



(c) Box-in-Box Coefficient Jump



(d) Sawtooth Coefficient Jump, when not visible, the lines for CℓAIR are underneath root-node.



(e) Laplace Problem with AMR.

Fig. 4: Various diffusion test problems, comparison of iterations and work-per-digit of accuracy for ℓAIR, CℓAIR, and root-node.

root-node, an AMG method known to be efficient for such problems. Our goal is not to find a faster solver for the 5-point Poisson operator, which would be difficult and of questionable value, but instead to verify the proposed solver. We have furthermore shown greater parameter insensitivity and dramatically

lower operator complexities when compared to ℓAIR for these problems. Finally, it is important to highlight that CℓAIR retains good convergence and low operator complexity across a range of diffusion problems, which has proven challenging for previous reduction-based approaches.

4.2. Advection-Diffusion Tests. We now turn our attention to nonsymmetric tests and consider the following two advection-diffusion problems.

2D Constant Direction Advection-Diffusion. Here the PDE is

$$-\alpha\nabla\cdot\nabla u + \mathbf{b}(x,y)\cdot\nabla u = f \quad \text{for } \Omega = [-1,1]^2, \tag{4.6a}$$

$$u = 0 \quad \text{on } \partial\Omega, \tag{4.6b}$$

where α is the diffusion constant and $\mathbf{b}(x,y)$ describes the advection. For this problem $\mathbf{b}(x,y) = [\sqrt{2/3}, \sqrt{1/3}]$ representing constant non-grid-aligned advection. The discretization is first-order upwinded discontinuous Galerkin (DG) for advection and the interior-penalty method for the DG discretization of diffusion. The discretization was generating by utilizing examples 14 (diffusion) and 9 (advection) with PyMFEM [2, 23]. For the case where $\alpha = 0$, the boundary conditions are modified slightly such that an outflow now occurs on the North and East walls ($u = 0$ is still prescribed on the West and South walls). We use $u = 0$ for the inflow, so that we may test our solver with a zero right-hand-side and random initial guess, as is commonly done to verify AMG solvers.

2D Recirculating Advection-Diffusion. Here the PDE and discretization is the same as for equations (4.6a)–(4.6b), except $\mathbf{b}(x,y) = [x(1-x)(2y-1), -(2x-1)(1-y)y]$, representing divergence-free recirculating advection about the origin. For $\alpha = 0$, this problem is ill-defined, thus we demonstrate numerical results over a range of diffusion constants (α), starting from a smallest diffusion constant $\alpha = 10^{-4}$ and going to $\alpha = 10.0$, so that we test the advective, mixed, and diffusive regimes.

4.2.1. Advection-Diffusion Results. Our goal here is to test the solver's robustness from the highly advective to highly diffusive regime and show improved performance and robustness relative to the current state ℓAIR. Similar to [20], we pre-scale the fine-grid matrices with the inverse of their diagonal block (block-size equals the DG element size of 4). Our solver parameters will remain fixed over these tests, and involve only minor changes to the symmetric parameters from Section 4.1. As the matrices are nonsymmetric, GMRES is accelerated with V(1,1)-cycles. The relaxation weight for postsmoothing with FFC-Jacobi is removed, as we no longer need symmetry for our preconditioner and this change slightly improves convergence for all methods.[6] The parameters for ℓAIR were changed slightly to use the second pass of Ruge-Stüben coarsening. The parameters for CℓAIR changed slightly with the strength-of-connection parameters becoming the same as for ℓAIR (0.25 for coarsening and 0.05 for interpolation). Additionally because of the nonsymmetry, P is generated separately from R using A and A^T, respectively. We also again consider two variants of CℓAIR, a high complexity version that uses first pass only Ruge-Stüben coarsening, and lower complexity versions that uses aggregation-based coarsening. The parameters for root-node changed similarly, where GMRES is now used for the energy-minimization when computing P and P is generated separately from R using A and A^T, respectively.

Figure 5a depicts work-per-digit for the 2D Constant Direction Advection problem with no diffusion ($\alpha = 0$). The purpose of this plot is to show that CℓAIR enjoys similar convergence to ℓAIR for this test problem, where we know that ℓAIR works well and is essentially the state-of-the-art [20].

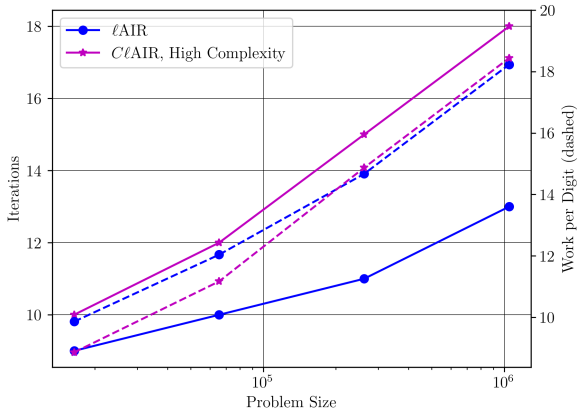
Figures 5c and 5d depict for all three solvers (with two different variants of CℓAIR) the work-per-digit of accuracy over a range of diffusion (α) values for the constant direction and recirculating test problems. The data points for root-node are omitted whenever the solver did not converge within 100 iterations (typically for small α values). We plainly see that CℓAIR has the most consistent performance in terms of work-per-digit accuracy across all regimes.

Table 2 highlights the operator complexity advantage of CℓAIR over ℓAIR for the constant advection case (the recirculating free case is similar). For the more advective cases (smaller α), the high complexity variant of CℓAIR, using first-pass only Ruge-Stüben coarsening, obtains lower operator complexities of roughly 15–30%, while for the more diffusive cases (larger α), the lower complexity variant of CℓAIR, using aggregation based coarsening, obtains operator complexities roughly 1.6x–2.5x smaller. In both settings, significant storage savings are achieved.

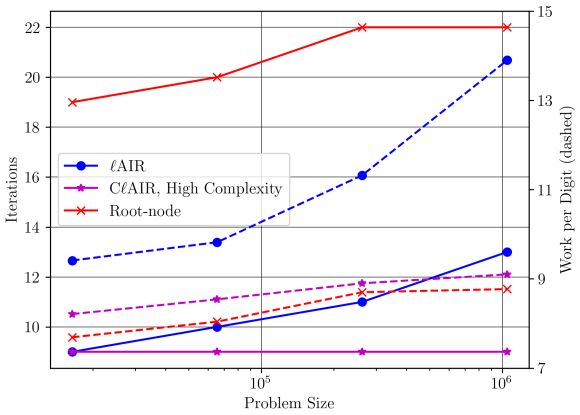
[6] The removal of the weight also more closely follows the reduction point-of-view in that post-smoothing should primarily solve the F-equations, so the slight improvement of convergence is not surprising.

α	10.0	1.0	0.1	0.01	0.001	0.0001	0.0
ℓAIR	3.19	3.19	3.20	3.19	3.70	3.19	4.28
CℓAIR, High Complexity	3.00	2.97	2.94	2.81	2.89	2.69	3.06
CℓAIR	1.28	1.29	1.29	1.96	DNC	DNC	DNC
Root-node	1.20	1.20	1.20	1.42	DNC	DNC	DNC

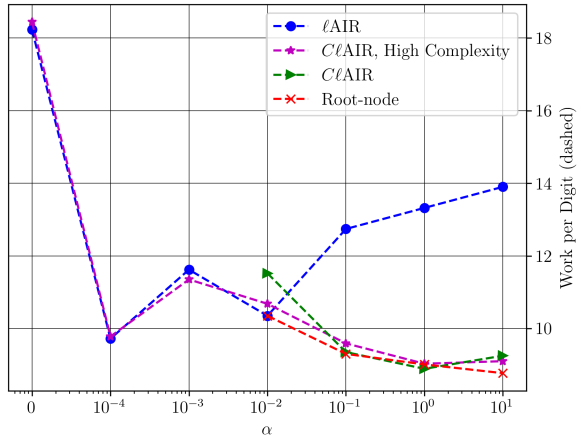
Table 2: Operator complexities of ℓAIR and CℓAIR for different diffusion α values and the constant advection test problem. Entries "DNC" indicate the solver did not converge within 100 iterations.



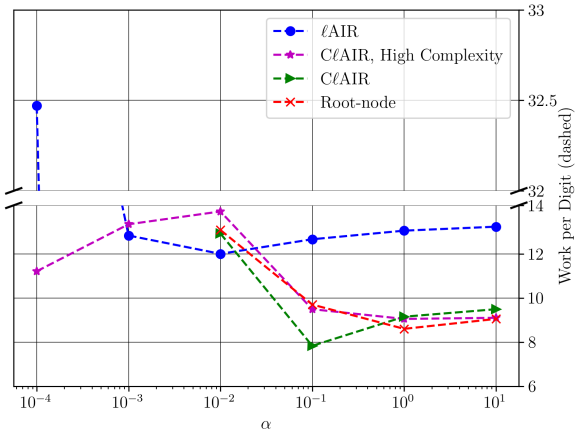
(a) Solver performance for 2D Constant Direction Advection, but no diffusion ($\alpha = 0$), comparing ℓAIR and CℓAIR. Note, root-node does not converge within 100 iterations for this problem and is not shown.



(b) Solver performance for 2D Constant Direction Advection with diffusion ($\alpha = 10$), comparing ℓAIR, CℓAIR and Root-node.



(c) Constant advection with varying diffusion



(d) Recirculating advection with varying diffusion

Fig. 5: Advection-diffusion problems, comparison of iterations, work-per-digit of accuracy and operator complexity for ℓAIR, CℓAIR and Root-node.

Remark 4.2. For these plots, our goal is to highlight the robustness (lack of need for tuning) for the CℓAIR solver. Thus, we searched for good general parameters for each solver and held them fixed over all tests. However, better parameters for individual problems and ℓAIR do exist. For instance, if the matrix is *not* block diagonally pre-scaled, then the solver iterations and work for ℓAIR in Figure 5b ($\alpha = 10$ case) improve and become 7, 8, 8, 9 and 10.35, 10.43, 11.60, 11.81, respectively, but with

a prohibitively large operator complexity of ~ 4.23. However with no pre-scaling, the ℓAIR then fails to converge within 100 iterations for the small α cases, e.g., Figure 5a. Another place where tuning is beneficial for ℓAIR is the smallest α value for the recirculating advection-diffusion problem ($\alpha = 10^{-4}$), where using first-pass only Ruge-Stüben coarsening lowers the iterations and work from 88 and 85.34 to 21 and 15.86, respectively. However, the use of first-pass only Ruge-Stüben coarsening then substantially degrades iterations and work-per-digit results for ℓAIR and other α values.

Remark 4.3. For a subset of the advection-diffusion problems considered in this section, Appendix A considers AMG approximation properties of CℓAIR and ℓAIR, with the results indicating that CℓAIR either maintains or improves on the approximation properties of ℓAIR.

In summary, the benefits of CℓAIR are as follows. The ℓAIR solver (i) requires more tuning for these problems than CℓAIR, (ii) requires more work-per-digit than CℓAIR for more diffusive problems (larger α values), and (iii) has significantly larger operator complexities.

5. Conclusion. In this paper, we developed a new type of reduction-based AMG that is suitable for solving nonsymmetric linear systems coming from the discretization of advection-diffusion PDEs. By combining techniques from ℓAIR that have been effective for advective problems, with energy minimization and root node techniques that are well suited for diffusion problems, we have developed an efficient method for solving advection-diffusion problems in a general setting – the solver is insensitive to varying contribution of the diffusive part in the PDE. An important distinction between our proposed solver and existing reduction based methods is that we take a column-wise approach to computing an approximation to ideal restriction and interpolation, which more naturally allows us to incorporate energy minimization and mode constraint techniques into the process. Our future work focuses on deriving a two-grid convergence theory for the proposed approach applied to nonsymmetric systems and to incorporate the idea of compatible relaxation into the CℓAIR coarsening process.

Appendix A. Classical AMG Weak and Strong Approximation Properties.

A.1. Approximation Properties in the Nonsymmetric Setting. We consider convergence of ℓAIR and CℓAIR based on classical multigrid weak and strong approximation properties. Targeting nonsymmetric problems, we consider the generalization of the A-norm as $\sqrt{A^*A}$ or $\sqrt{AA^*}$ [7, 21]. For a nonsingular matrix $A \in \mathbb{C}^{n\times n}$, consider the singular value decomposition (SVD) given by $A = U\Sigma V^*$ where the singular values are $0 < \sigma_1 \leq \sigma_2 \leq ... \leq \sigma_n$. Then $\sqrt{A^*A} = V\Sigma V^* = VU^*U\Sigma V^* = QA$ where $Q = VU^*$. In a similar manner, we can also obtain $\sqrt{AA^*} = U\Sigma U^* = AQ$. Since $\sqrt{A^*A}$ or $\sqrt{AA^*}$ are SPD matrices, therefore we can still consider classical AMG approximation properties with respect to the SPD matrices QA and AQ corresponding to the right and left singular vectors, respectively. Such approximation properties measure, in a sense, how effective the coarse spaces are at capturing the near nullspace of the operator.

For several test problems, we numerically evaluate approximation properties for ℓAIR and CℓAIR by making use of the generalized fractional approximation property (FAP) [21]. The FAP of a transfer operator $T \in \mathbb{R}^{n\times n_c}$ is with respect to the SPD matrix $\mathcal{A}$, with powers $\beta, \eta \geq 0$ and constant $K_{T,\beta,\eta}$. Specifically, T is said to have a FAP if for every fine grid vector, $\mathbf{v}$, there exists a coarse grid vector, $\mathbf{v}_c$, such that

$$||\mathbf{v} - T\mathbf{v}_c||^2_{\mathcal{A}^\eta} \leq \frac{K_{T,\beta,\eta}}{||\mathcal{A}||^{2\beta-\eta}} \langle \mathcal{A}^{2\beta}\mathbf{v}, \mathbf{v}\rangle, \tag{A.1}$$

where $\mathcal{A} = QA$ ($T = P$) or $\mathcal{A} = AQ$ ($T = R^*$). The classical multigrid weak approximation property (WAP) is a FAP$(1/2, 0)$, that is

$$||\mathbf{v} - T\mathbf{v}_c||^2_2 \leq \frac{K_{T,1/2,0}}{||\mathcal{A}||} \langle \mathcal{A}\mathbf{v}, \mathbf{v}\rangle. \tag{A.2}$$

Further, the classical multigrid strong approximation property (SAP) is a FAP(1,1), that is

$$||\mathbf{v} - T\mathbf{v}_c||^2_{\mathcal{A}} \leq \frac{K_{T,1,1}}{||\mathcal{A}||} \langle \mathcal{A}^2\mathbf{v}, \mathbf{v}\rangle. \tag{A.3}$$

For a given vector $\mathbf{v}$, we compute the approximation constant $K_{T,\beta,\eta}(\mathbf{v})$ with

$$K_{T,\beta,\eta}(\mathbf{v}) = \frac{||\mathcal{A}||^{2\beta-\eta}}{||\mathbf{v}||^2_{\mathcal{A}^{2\beta}}} \min_{\mathbf{v}_c} ||\mathbf{v} - T\mathbf{v}_c||^2_{\mathcal{A}^\eta}. \tag{A.4}$$

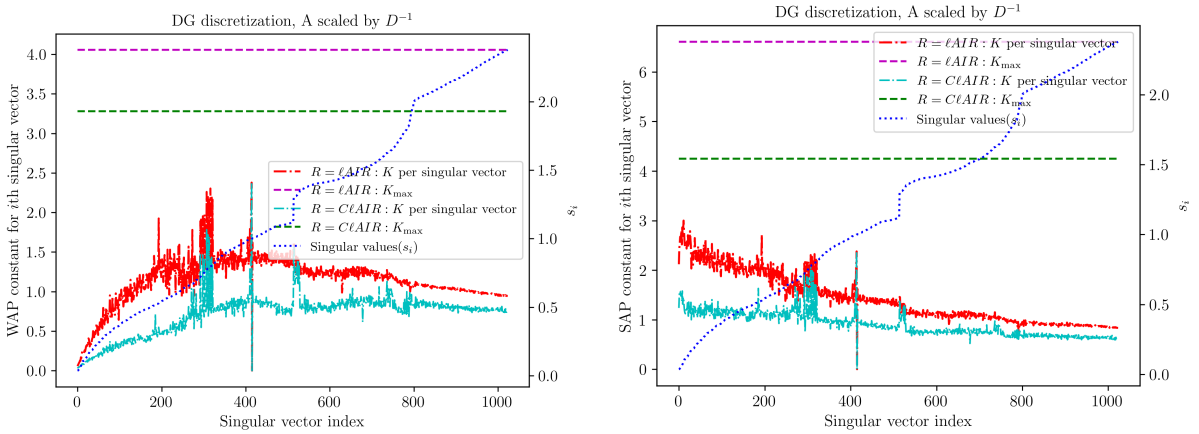
Let Π_η denote the $\mathcal{A}^\eta$-orthogonal projection onto the range of T, $\Pi_\eta = T(T^*\mathcal{A}^\eta T)^{-1}T^*\mathcal{A}^\eta$. Substituting into (A.4) we obtain,

$$K_{T,\beta,\eta}(\mathbf{v}) = \frac{||\mathcal{A}||^{2\beta-\eta}}{||\mathbf{v}||^2_{\mathcal{A}^{2\beta}}}\ ||(I-\Pi_\eta)\mathbf{v}||^2_{\mathcal{A}^\eta}. \tag{A.5}$$

To compute the approximation constant $K_{\max}$ that holds for all fine grid vectors $\mathbf{v}$, we take maximum of the above expression over all $\mathbf{v}$, which leads to

$$K_{\max} = \max_{\mathbf{v}\neq 0}\ K_{T,\beta,\eta}(\mathbf{v}) = ||\mathcal{A}||^{2\beta-\eta}\ ||\mathcal{A}^{\eta/2}(I-\Pi_\eta)\mathcal{A}^{-\beta}||_2^2. \tag{A.6}$$

A.2. Numerical Tests. To complement the numerical results in the main text, we now measure classical AMG approximation property constants for both ℓAIR and CℓAIR from the purely advective to diffusion dominated case. Specifically, we consider the 2D constant direction advection problem (4.6a)–(4.6b) for various diffusion coefficients α. For numerical tests, a 32×32 size spatial domain is considered, resulting in 1024 total DOFs. In the following numerical tests, we consider the approximation properties of the restriction operators from both ℓAIR and CℓAIR.



(a) WAP for Constant Advection with zero diffusion ($\alpha = 0$) where constraint vector $B = \mathbf{1}$ is presmoothed with 5 iterations of CFF-weighted-Jacobi.

(b) SAP for Constant Advection with zero diffusion ($\alpha = 0$) where constraint vector $B = \mathbf{1}$ is presmoothed with 5 iterations of CFF-weighted-Jacobi.

Fig. 6: WAP and SAP constants for the restriction operators of ℓAIR and CℓAIR for the constant advection with zero diffusion ($\alpha = 0$) problem. Five (5) iterations of CFF-weighted-Jacobi relaxation has been used to improve the mode constraint vector $B = \mathbf{1}$ in CℓAIR. Singular values are shown in the dotted blue line and are associated with the right vertical axis, and the dot-dashed lines show the approximation constant for each of the left singular vectors of A. Horizontal dashed lines show the approximation constant for ℓAIR and CℓAIR that holds for all vectors.

Figure 6 shows the WAP (FAP$(1/2, 0)$) and SAP (FAP$(1, 1)$) approximation constants for the left singular vectors of A. Here, 5 iterations of CFF-weighted-Jacobi relaxation has been used to improve the mode constraint vector $B = \mathbf{1}$. From Figure 6a and 6b, we see that CℓAIR demonstrates moderately smaller (i.e., better) WAP and SAP constants than ℓAIR. Since ℓAIR is primarily designed for such purely advective problems these results indicate that CℓAIR is also effective for advection problems, with similar convergence properties. This is indeed what we see in Figures 5a and 5c. This motivates us to further study approximation properties of CℓAIR for diffusion dominated problems.

Next, we consider a diffusion dominated case by setting the diffusion coefficient to be $\alpha = 10$. First, as previously, we use 5 iterations of a CFF-weighted-Jacobi smoother to improve the mode constraint vector $B = \mathbf{1}$. Carefully investigating the results in Figure 7a, we observe that WAP constant for the new method CℓAIR is almost a factor of two smaller than that for ℓAIR. Now if we compare the SAP

constants in Figure 7b, $K_{\max}$ is smaller for CℓAIR than for ℓAIR, although the constants are very large for both solvers (184 for CℓAIR and 2246 for ℓAIR). Somewhat surprisingly, while the SAP constant for CℓAIR in Figure 7b is quite large, we find that the iteration counts of the solver are mesh independent (as we have seen in Figure 5b for the $\alpha = 10$ case). In this case, we suspect that the Krylov method is able to account for what the solver is lacking. We note that the significantly larger SAP constants for ℓAIR are consistent with the poorer scalability of ℓAIR seen in Figure 5b for the $\alpha = 10$ case. A key advantage of CℓAIR is that it allows the flexibility to improve the approximation properties of the restriction operator by employing a suitable relaxation scheme for smoothing the mode constraint vector (such as weighted CFF-Jacobi). This type of flexibility is not available in ℓAIR. Therefore, further improvement of CℓAIR's SAP constant can be obtained by increasing the number of iterations applied to B. In our experiments, we find that if the mode constraint vector $B = \mathbf{1}$ is presmoothed with 25 iterations of CFF-weighted-Jacobi (instead of 5 iterations), the SAP approximation constant for CℓAIR decreases further to $K_{\max} = 77$ (Figure 7d). However, we do not find that these extra iterations lead to a meaningful improvement in practice for CℓAIR convergence on the tested problems.

Acknowledgments. Los Alamos National Laboratory Report LA-UR-23-27101.

REFERENCES

[1] Raymond E Alcouffe, Achi Brandt, Joel E Dendy, Jr, and James W Painter. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. *SIAM Journal on Scientific and Statistical Computing*, 2(4):430–454, 1981.

[2] Robert Anderson, Julian Andrej, Andrew Barker, Jamie Bramwell, Jean-Sylvain Camier, Jakub Cerveny, Veselin Dobrev, Yohann Dudouit, Aaron Fisher, Tzanio Kolev, Will Pazner, Mark Stowell, Vladimir Tomov, Ido Akkerman, Johann Dahm, David Medina, and Stefano Zampini. MFEM: A modular finite element methods library. *Computers & Mathematics with Applications*, 81:42–74, 2021.

[3] Nathan Bell, Luke N. Olson, Jacob Schroder, and Ben S. Southworth. PyAMG: Algebraic Multigrid Solvers in Python. *Journal of Open Source Software*, 2022. submitted.

[4] A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations. Technical report, Colorado State University, Fort Collins, Colorado, 1983.

[5] Achi Brandt, James Brannick, Karsten Kahl, and Ira Livshits. Algebraic distance for anisotropic diffusion problems: Multilevel results. *Electronic Transactions on Numerical Analysis*, 44:472–496, 2015.

[6] J. Brannick and L.Zikatanov. Algebraic multigrid methods based on compatible relaxation and energy minimization. In *Proceedings of the 16th International Conference on Domain Decomposition Methods*, volume 55 of *Lecture Notes in Computational Science and Engineering*, pages 15–26. Springer, 2007.

[7] Marian Brezina, T Manteuffel, S MCormick, J Ruge, and Geoffrey Sanders. Towards adaptive smoothed aggregation (αSA) for nonsymmetric problems. *SIAM Journal on Scientific Computing*, 32(1):14–39, 2010.

[8] S Dargaville, RP Smedley-Stevenson, PN Smith, and CC Pain. AIR multigrid with GMRES polynomials (AIRG) and additive preconditioners for Boltzmann transport. *arXiv preprint arXiv:2301.05521*, 2023.

[9] Hans De Sterck, Robert D Falgout, Stephanie Friedhoff, Oliver A Krzysik, and Scott P MacLachlan. Optimizing multigrid reduction-in-time and parareal coarse-grid operators for linear advection. *Numerical Linear Algebra with Applications*, 28(4):e2367, 2021.

[10] VA Dobrev, Tz Kolev, N Anders Petersson, and Jacob B Schroder. Two-level convergence theory for multigrid reduction in time (MGRIT). *SIAM Journal on Scientific Computing*, 39(5):S501–S527, 2017.

[11] Robert D Falgout, Stephanie Friedhoff, Tz V Kolev, Scott P MacLachlan, and Jacob B Schroder. Parallel time integration with multigrid. *SIAM Journal on Scientific Computing*, 36(6):C635–C661, 2014.

[12] Robert D Falgout and Panayot S Vassilevski. On generalizing the algebraic multigrid framework. *SIAM Journal on Numerical Analysis*, 42(4):1669–1693, 2004.

[13] Andreas Frommer, Karsten Kahl, Scott P. MacLachlan, Ludmil T. Zikatanov, and James Brannick. Adaptive reduction-based multigrid for nearly singular and highly disordered physical systems. *Electronic Transactions on Numerical Analysis*, 37:276–295, 2010.

[14] Nicholas IM Gould, Mary E Hribar, and Jorge Nocedal. On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM Journal on Scientific Computing*, 23(4):1376–1395, 2001.

[15] Karsten Kahl and Matthias Rottmann. Least angle regression coarsening in bootstrap algebraic multigrid. *SIAM Journal on Scientific Computing*, 40(6):A3928–A3954, 2018.

[16] Scott MacLachlan, Tom Manteuffel, and Steve McCormick. Adaptive reduction-based AMG. *Numerical Linear Algebra with Applications*, 13(8):599–620, 2006.

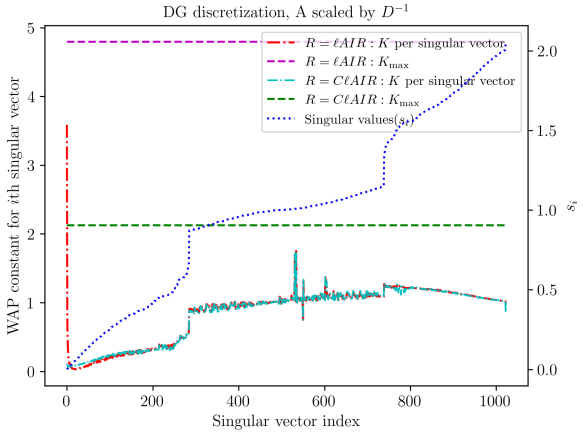
[17] Jan Mandel, Marian Brezina, and Petr Vanek. Energy optimization of algebraic multigrid bases. Technical report, University of Colorado Denver, 1998.

[18] Thomas A Manteuffel, Steffen Münzenmaier, John Ruge, and Ben Southworth. Nonsymmetric reduction-based algebraic multigrid. *SIAM Journal on Scientific Computing*, 41(5):S242–S268, 2019.

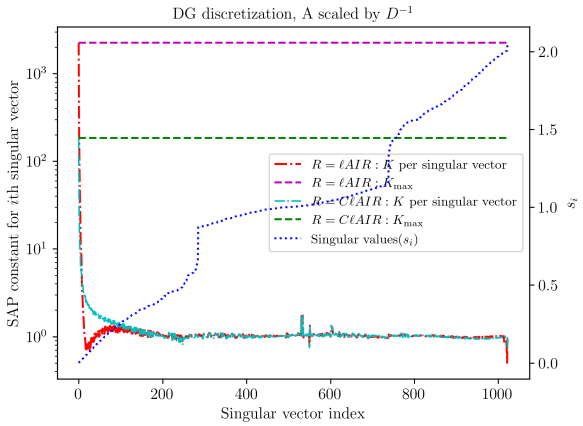
[19] Thomas A Manteuffel, Luke N Olson, Jacob B Schroder, and Ben S Southworth. A root-node–based algebraic multigrid method. *SIAM Journal on Scientific Computing*, 39(5):S723–S756, 2017.

[20] Thomas A Manteuffel, John Ruge, and Ben S Southworth. Nonsymmetric algebraic multigrid based on local approximate ideal restriction (ℓAIR). *SIAM Journal on Scientific Computing*, 40(6):A4105–A4130, 2018.

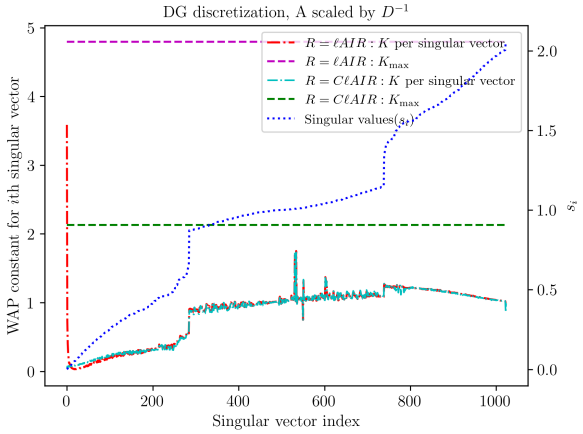
[21] Tom Manteuffel and Ben S Southworth. Convergence in norm of nonsymmetric algebraic multigrid. *SIAM Journal*



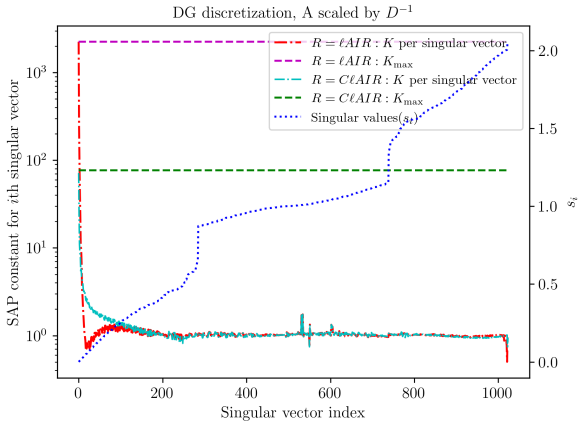
(a) WAP for Constant Advection with added diffusion ($\alpha = 10$) where constraint vector $B = \mathbf{1}$ is presmoothed with 5 iterations of CFF-weighted-Jacobi.



(b) SAP for Constant Advection with added diffusion ($\alpha = 10$) where constraint vector $B = \mathbf{1}$ is presmoothed with 5 iterations of CFF-weighted-Jacobi.



(c) WAP for Constant Advection with added diffusion ($\alpha = 10$) where constraint vector $B = \mathbf{1}$ is presmoothed with 25 iterations of CFF-weighted-Jacobi.



(d) SAP for Constant Advection with added diffusion ($\alpha = 10$) where constraint vector $B = \mathbf{1}$ is presmoothed with 25 iterations of CFF-weighted-Jacobi.

Fig. 7: WAP and SAP constants for the restriction operators of ℓAIR and CℓAIR for the constant advection with added diffusion ($\alpha = 10$) problem. Different number of iterations (5 and then 25) of CFF-weighted-Jacobi relaxation has been used to improve the mode constraint vector $B = \mathbf{1}$ in CℓAIR. Singular values are shown in the dotted blue line and are associated with the right vertical axis, and the dot-dashed lines show the approximation constant for each of the left singular vectors of A. Horizontal dashed lines show the approximation constant for ℓAIR and CℓAIR that holds for all vectors.

on Scientific Computing, 41(5):S269–S296, 2019.

[22] Luke N. Olson, Jacob B. Schroder, and Raymond S. Tuminaro. A general interpolation strategy for algebraic multigrid using energy minimization. *SIAM Journal on Scientific Computing*, 33(2):966–991, 2011.

[23] PyMFEM. https://github.com/mfem/PyMFEM/.

[24] J. W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. F. McCormick, editor, *Multigrid Methods*, volume 3 of *Frontiers in Applied Mathematics*, pages 73–130. SIAM, Philadelphia, PA, 1987.

[25] J. B. Schroder. Smoothed aggregation solvers for anisotropic diffusion. *Numer. Linear Algebra Appl.*, 19:296–312, 2012.

[26] Abdullah Ali Sivas, Ben S Southworth, and Sander Rhebergen. AIR algebraic multigrid for a space-time hybridizable discontinuous galerkin discretization of advection (-diffusion). *SIAM Journal on Scientific Computing*, 43(5):A3393–A3416, 2021.

[27] Ben S Southworth. Necessary conditions and tight two-level convergence bounds for parareal and multigrid reduction in time. *SIAM Journal on Matrix Analysis and Applications*, 40(2):564–608, 2019.

[28] Ben S Southworth, Oliver Krzysik, and Will Pazner. Fast Solution of Fully Implicit Runge –Kutta and Discontinuous Galerkin in Time for Numerical PDEs, Part II: Nonlinearities and DAEs. *SIAM Journal on Scientific Computing*, 44(2):A636–A663, 2022.

[29] Ben S Southworth, Wayne Mitchell, Andreas Hessenthaler, and Federico Danieli. Tight two-level convergence of linear parareal and MGRIT: Extensions and implications in practice. In *Parallel-in-Time Integration Methods: 9th Parallel-in-Time Workshop, June 8–12, 2020*, pages 1–31. Springer, 2021.

[30] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid based on smoothed aggregation for second and fourth order problems. *Computing*, 56:179–196, 1996.

[31] W. L. Wan, Tony F. Chan, and Barry Smith. An energy-minimizing interpolation for robust multigrid methods. *SIAM J. Sci. Comp.*, 21(4):1632–1649, 2000.

[32] Tareq Zaman, Nicolas Nytko, Ali Taghibakhshi, Scott MacLachlan, Luke Olson, and Matthew West. Generalizing reduction-based algebraic multigrid. *Numerical Linear Algebra with Applications*, 31(3):e2543, 2023.

[33] Olgierd Cecil Zienkiewicz and Jian Zhong Zhu. The superconvergent patch recovery and a posteriori error estimates. part 2: Error estimates and adaptivity. *International Journal for Numerical Methods in Engineering*, 33(7):1365–1382, 1992.