SIAM J. SCI. COMPUT.
Vol. 41, No. 5, pp. S242–S268

NONSYMMETRIC REDUCTION-BASED ALGEBRAIC MULTIGRID[*]

THOMAS A. MANTEUFFEL[†], STEFFEN MÜNZENMAIER[‡], JOHN RUGE[†], AND BEN SOUTHWORTH[†]

Abstract. Algebraic multigrid (AMG) is often an effective solver for symmetric positive definite (SPD) linear systems resulting from the discretization of general elliptic PDEs or the spatial discretization of parabolic PDEs. However, convergence theory and most variations of AMG rely on A being SPD. Hyperbolic PDEs, which arise often in large-scale scientific simulations, remain a challenge for AMG, as well as other fast linear solvers, in part because the resulting linear systems are often highly nonsymmetric. Here, a novel convergence framework is developed for nonsymmetric, reduction-based AMG, and sufficient conditions derived for ℓ^2-convergence of error and residual. In particular, classical multigrid approximation properties are connected with reduction-based measures to develop a robust framework for nonsymmetric, reduction-based AMG. Matrices with block-triangular structure are then recognized as being amenable to reduction-type algorithms, and a reduction-based AMG method is developed for upwind discretizations of hyperbolic PDEs, based on the concept of a Neumann approximation to ideal restriction (nAIR). nAIR can be seen as a variation of local AIR (ℓAIR) introduced in previous work, specifically targeting matrices with triangular structure. Although less versatile than ℓAIR, setup times for nAIR can be substantially faster for problems with high connectivity. nAIR is shown to be an effective and scalable solver of steady state transport for discontinuous, upwind discretizations, with unstructured meshes, and up to 6th-order finite elements, offering a significant improvement over existing AMG methods. nAIR is also shown to be effective on several classes of "nearly triangular" matrices resulting from curvilinear finite elements and artificial diffusion.

Key words. multigrid, nonsymmetric, hyperbolic, algebraic, AMG, reduction

AMS subject classifications. 65F08, 65F10, 65N22, 65N55, 35L02

DOI. 10.1137/18M1193761

1. Introduction.

Solving large, sparse linear systems is fundamental to the numerical solution of partial differential equations (PDEs). For high-dimensional PDEs, even a moderate resolution of the discrete problem can lead to enormous problem sizes, which require highly efficient, parallel solvers. Ultimately, it is important that a solver is both *algorithmically scalable* (*fast*), with a cost in floating point operations linear or near-linear with the number of unknowns, and *scalable in parallel* (*scalable*), where these operations can be efficiently distributed across many processors. For symmetric positive definite (SPD) matrices, such as those that arise in the discretization of elliptic PDEs or the spatial discretization of a parabolic PDE in time, a number of fast, scalable, iterative, or direct methods have been developed. However, there remains a lack of fast, scalable solvers for the highly nonsymmetric matrices that arise in the discretization of hyperbolic PDEs and full space-time discretizations of general

[*] Received by the editors June 12, 2018; accepted for publication (in revised form) August 20, 2019; published electronically October 29, 2019.
https://doi.org/10.1137/18M1193761

Funding: This work was supported by the U.S. Department of Defense, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. This work was performed under the auspices of the U.S. Department of Energy under grants (SC) DE-FC02-03ER25574 and (NNSA) DE-NA0002376, and the Lawrence Livermore National Laboratory under contract B614452.

[†] Department of Applied Mathematics, University of Colorado, Boulder, Boulder, CO 80309-0526 (tmanteuf@colorado.edu, John.Ruge@colorado.edu, ben.s.southworth@gmail.com).

[‡] Fakultät für Mathematik, Universität Duisburg-Essen, 47057 Duisburg, Germany (steffen.muenzenmaier@uni-due.de).

PDEs. A subset of such highly nonsymmetric matrices are block-triangular matrices, with blocks that are small enough to invert directly, which is one focus of this work.

For PDEs of elliptic type, algebraic multigrid (AMG) is among the fastest class of linear solvers. When applicable, AMG converges in linear complexity with the number of degrees-of-freedom (DOFs) and scales in parallel like $O(\log_2(P))$, up to hundreds of thousands of processors, P [6]. Originally, AMG was designed for essentially SPD linear systems, and convergence of AMG is relatively well understood in the SPD setting [10, 11, 26, 60, 63]. Nonsymmetric matrices pose unique difficulties for AMG in both theory and practice. In particular, coarse-grid correction, a fundamental part of AMG's fast convergence, is generally not a contraction in the nonsymmetric setting, meaning that coarse-grid correction can increase error. There have been efforts to develop AMG theory and methods in the nonsymmetric setting [13, 36, 39, 41, 45, 45, 46, 47, 56, 61]. However, the theoretical understanding of nonsymmetric AMG remains limited, with very few convergence bounds proved in norm, and there has yet to be a robust and scalable AMG solver for highly nonsymmetric problems.

For the most part, previous work on nonsymmetric AMG has appealed to traditional AMG thought, where coarse-grid correction captures (right) singular vectors associated with small singular values and relaxation attenuates error associated with large singular values. Here, we take a different, reduction-based approach, appealing to the premise that certain "ideal" restriction and interpolation operators can lead to a reduction-based (nonsymmetric) AMG method. Although a reduction-based solver is not a new concept, here we recognize (i) an important class of linear systems for which reduction can be highly effective, and (ii) develop a theoretical framework explaining the convergence of nonsymmetric reduction-based AMG in norm.

Background on sparse triangular systems, reduction, and reduction-based AMG is given in section 2. A novel convergence framework is then developed for nonsymmetric, reduction-based AMG in section 3, including sufficient conditions for ℓ^2-convergence of error and residual, and a formal connection of reduction-based AMG to classical multigrid approximation properties. Section 4 then recognizes block-triangular and near-triangular matrices to be well suited for reduction-based AMG, and discusses such operators in the context of discontinuous discretizations of advection. In particular, for hyperbolic-type PDEs, a Neumann approximation to the ideal restriction operator (nAIR) provides an accurate, sparse approximation, and an AMG algorithm referred to as nAIR is developed based on this principle. This is a similar approach to that developed in [58] (ℓAIR), but nAIR can offer setup-times that are orders of magnitude faster than ℓAIR in some cases. Steady state transport is used as a model problem, which arises in large-scale simulation of neutron and radiation transport [1, 5, 18, 31, 50]. Results in section 6 show that nAIR outperforms current state-of-the-art methods, and is able to attain an order-of-magnitude reduction in residual in only 1–2 iterations, even for high-order discretizations on unstructured grids.

2. Reduction-based algebraic multigrid.

2.1. Triangular matrices and parallelism.

Sparse matrices with triangular structure arise in a number of interesting settings. In contrast to elliptic and parabolic PDEs, the solution of hyperbolic PDEs lies on characteristic curves, and the solution at any point depends only on the solution upwind along its given characteristic. This allows for very steep gradients or "fronts" to propagate through the domain. Due to such behavior, discontinuous, upwind discretizations are a natural approach to discretizing many hyperbolic PDEs [14, 43, 44, 51]. For a fully upwinded discretization, the resulting matrix has a block-triangular structure in some (although potentially unknown) ordering. Implicit time-stepping schemes or steady state solutions then require the solution of such linear systems. There has also been growing interest in parallel-in-time solvers. Although most work on these has been geometric in nature, one can also consider algebraically solving the sparse matrix equations associated with a full space-time discretization. Such discretizations using an explicit time-stepping scheme (for any PDE) result in triangular matrices, and an implicit time-stepping scheme coupled with an upwind discretization of a hyperbolic PDE also leads to a block triangular matrix.

Solving linear systems with block-triangular structure is easy in serial using a forward or backward solve. However, there are cases where a block-triangular solve arises in the parallel setting. In particular, any time an upwind advective discretization is either (i) part of a larger PDE discretization that cannot be stored on a small number of processors or (ii) coupled to other variables that are not inverted easily and require parallel preconditioners, a triangular-type solve is necessary in parallel. Scheduling algorithms have been developed for sparse matrices that can add some level of parallelism to this process, but such algorithms are primarily relevant for shared memory environments [2, 4, 34, 35], and, from the perspective of simulation of PDEs, efficiency in a distributed-memory environment is fundamental. In the simplest case of a perfectly structured processor grid of squares/cubes over some d-dimensional domain and a fixed, constant direction of flow, a forward solve in parallel scales like $O(P^{1/d})$ for P processors [5]. However, for nonconstant flow or nonuniform processor grids with respect to the flow and domain, this convergence can suffer. In fact, for *any* processor configuration over a given domain, it is straightforward to construct a velocity field for advection such that a distributed forward solve takes P communication steps to complete. Even if each step of this process is fast, linear or even square-root scaling in P is poor parallel performance.

Iterative-type methods offer an alternative (potentially) more amenable to parallelism. However, Krylov and most other traditional iterative-type methods are generally either divergent or too slow (with respect to convergence) to be considered a tractable approach, and solving large, sparse, block-triangular linear systems in (distributed-memory) parallel environments remains a difficult task. Geometric multigrid has been applied to hyperbolic PDEs and upwind discretizations using the well-known line-smoother approach [3, 49, 64], but such an approach requires a fixed/known direction over which to relax and has limits in terms of parallel scalability. When considering time-dependent hyperbolic PDEs, explicit time-stepping schemes can be used to avoid solving linear systems. But, explicit schemes suffer from stability constraints, such as the Courant–Friedrichs–Lewy (CFL) condition, which often require extremely small time steps, a process that is sequential and can limit performance in the parallel setting. AMG is known to scale like $O(\log P)$ in parallel, and an AMG solver that is effective on block triangular and near-triangular matrices would overcome many of the aforementioned difficulties.

2.2. Reduction-based AMG.

Reduction consists of solving a problem by equivalently solving multiple smaller problems. In terms of triangular systems, a direct forward or backward solve is a reduction algorithm: starting with a system of size $n \times n$, eliminate one DOF, and reduce the problem to size $(n-1) \times (n-1)$. Although this is a sequential algorithm, it suggests that reduction is an effective and tractable approach for triangular systems, which we demonstrate in this paper.

Given an $n \times n$ matrix A, suppose the n DOFs are partitioned into n_c C-points and n_f F-points. For explanation, assume that the unknowns have been ordered with the F-points followed by the C-points. In this context, a well-known example of reduction is a 2×2 block LDU decomposition:

$$\begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix}^{-1} = \begin{bmatrix} I & -A_{ff}^{-1}A_{fc} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{ff}^{-1} & 0 \\ 0 & \mathcal{K}_A^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{cf}A_{ff}^{-1} & I \end{bmatrix}. \tag{1}$$

Here, solving $A\mathbf{x} = \mathbf{b}$ is reduced to solving two systems, $A_{ff} \in \mathbb{R}^{n_F \times n_F}$ and the Schur complement, $\mathcal{K}_A := A_{cc} - A_{cf}A_{ff}^{-1}A_{fc} \in \mathbb{R}^{n_C \times n_C}$. LDU decompositions assume that the action of A_{ff}^{-1} is available to compute the action of the first and third matrix blocks in (1), while in practice it is typically not. However, approximating such a decomposition is the goal behind numerous preconditioners. In fact, a two-level reduction-based AMG method can be posed as a variant of a block LDU decomposition, which is what we develop here by algebraically approximating the operators that lead to reduction in (1): $A_{cf}A_{ff}^{-1}$ and $A_{ff}^{-1}A_{fc}$.

AMG applied to the linear system $A\mathbf{x} = \mathbf{b}$, $A \in \mathbb{R}^{n \times n}$, consists of two processes: relaxation and coarse-grid correction, designed to be complementary in the sense that they effectively reduce error associated with different parts of the spectrum of A. Relaxation often takes the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + M^{-1}(\mathbf{b} - A\mathbf{x}_k), \tag{2}$$

where M^{-1} is some approximation to A^{-1} such that the action of M^{-1} can be easily computed. Coarse-grid correction typically takes the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + P(RAP)^{-1}R(\mathbf{b} - A\mathbf{x}_k), \tag{3}$$

where $P \in \mathbb{R}^{n \times n_c}$ and $R \in \mathbb{R}^{n_c \times n}$ are interpolation and restriction operators, respectively, between $\mathbb{R}^n$ and the next coarser grid in the AMG hierarchy, $\mathbb{R}^{n_c}$. Denote the projection $\Pi := P(RAP)^{-1}RA$. A two-level $V(1,1)$-cycle is then given by combining coarse-grid correction in (3) with pre- and postrelaxation steps as in (2), resulting in a two-grid error propagation operator of the form

$$E_{TG} = (I - M_{\text{post}}^{-1}A)(I - \Pi)(I - M_{\text{pre}}^{-1}A).$$

A classical AMG approach is used here, where a CF-splitting of DOFs defines the coarse grid [10, 54]. Operators A, P, and R can then be written in block form:

$$A = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix}, \qquad P = \begin{bmatrix} W \\ I \end{bmatrix}, \qquad R = \begin{bmatrix} Z & I \end{bmatrix}, \tag{4}$$

where $W \in \mathbb{R}^{n_f \times n_c}$ interpolates to F-points via linear combinations of coarse-grid DOFs, and $Z \in \mathbb{R}^{n_c \times n_f}$ restricts F-point residuals. Note that (4) implicitly assumes the same CF-splitting for R and P, although the sparsity patterns for nonzero elements of Z^T and W may be different. For notation, denote the coarse-grid operator:[1]

$$\mathcal{K} : = RAP = ZA_{ff}W + A_{cf}W + ZA_{fc} + A_{cc}. \tag{5}$$

Define the "ideal restriction" and "ideal interpolation" operators as

$$R_{\text{ideal}} := \begin{bmatrix} -A_{cf}A_{ff}^{-1} & I \end{bmatrix}, \qquad P_{ideal} := \begin{bmatrix} -A_{ff}^{-1}A_{fc} \\ I \end{bmatrix},$$

[1] $\mathcal{K}$ is used to denote the coarse-grid operator instead of the traditional notation, A_c, to avoid confusion with subscripts denoting C-points.

These operators define the LDU reduction in (1). Note that $\mathcal{K} := \mathcal{K}_A$, with $P = P_{\text{ideal}}$ *or* $R = R_{\text{ideal}}$ independent of one another. Ideal interpolation has been explored in the context of reduction-based geometric multigrid methods [22, 23, 28, 53], and is well-motivated under classical AMG theory for SPD matrices, where it is optimal in a certain sense with respect to two-grid convergence [26, 63]. Ideal restriction was also discussed in the context of reduction in [26] and, in [58, section 2.3], shown to be the unique restriction operator that gives an exact coarse-grid correction at C-points, independent of interpolation. This result, along with a corollary on ideal interpolation [58, section 2.2], are summarized in the following results.

Lemma 1 (ideal restriction). *For a given CF-splitting, assume that A_{ff} is nonsingular, and let A, P, and R take the block form as given in* (4). *Then, an exact coarse-grid correction at C-points is attained for all* $\mathbf{e}$ *if and only if* $R = R_{\text{ideal}}$. *Furthermore, the error in coarse-grid correction is given by*

$$(I - \Pi)\mathbf{e} = \begin{bmatrix} \mathbf{e}_f - W\mathbf{e}_c \\ \mathbf{0} \end{bmatrix}. \tag{6}$$

A coarse-grid correction using R_{ideal} *followed by an exact solve on F-points results in an exact two-grid solver, independent of* W.

Corollary 2 (ideal interpolation). *For a given CF-splitting, assume that A_{ff} is nonsingular, and let A, P, and R take the block form as given in* (4). *Then, an exact solve on F-points, followed by a coarse-grid correction using $P := P_{ideal}$, yields an exact two-level solver, independent of Z.*

Thus, in the nonsymmetric setting, ideal transfer operators are "ideal" in a reduction sense; when coupled with an exact solve on F-points, R_{ideal} and P_{ideal} each lead to an exact two-level method, independent of the accompanying interpolation and restriction operators, respectively. Note that the order of solving the coarse- and fine-grid problems is important, to be an exact reduction, the F-point solve must *follow* coarse-grid correction with R_{ideal}, while the F-point solve must *precede* coarse-grid correction with P_{ideal}.

2.3. Relation to existing methods.

Schur-complement preconditioning and reduction-based solvers are not new to the literature. Numerous algorithms have been based on a block LDU decomposition (1) and approximate Schur complement (for example, [8, 17, 38, 41, 42, 45, 55]). Reduction has also been studied in the multigrid and AMG context, originally in the geometric setting [28, 53], more recently algebraically [12, 37], and also as the basis for the multigrid reduction-in-time (MGRIT) parallel-in-time method [21, 22, 23]. MGRIT is designed for nonsymmetric problems, but is geometric in nature and relies on the very specific matrix structure that arises in time integration, more or less a block one-dimensional (1D) advection problem. The AMG developments in [12, 37] are fully algebraic and reduction-based, but assume a Galerkin coarse grid, meaning restriction is defined as $R = P^T$. For the highly nonsymmetric systems considered here, this is typically not a good choice [40, 58], motivating a Petrov–Galerkin method, where $R \neq P^T$. Unfortunately, choosing $R \neq P^T$ also introduces new difficulties, because if R and P are not "compatible" in some sense, the norm of coarse-grid correction can be $\gg 1$ [40, 58], leading to a divergent method.

Approximating the ideal restriction or interpolation operators has also been considered in other (nonreduction-based) AMG methods, including [39, 48, 61]. Also motivated in a block-LDU sense, ideal restriction and interpolation are approximated in [61] by performing a constrained minimization over a fixed sparsity pattern for R and P. A similar constrained minimization approach for nonsymmetric systems was used in [39, 48], where ideal operators of A^*A and AA^* for P and R, respectively, are approximated using a constrained minimization. In these cases, the solvers appeal to more classical convergence theory by enforcing constraints to interpolate certain (known) vectors associated with small singular values exactly.

The AIR algorithm, developed here and in [58], takes a somewhat converse approach. Building on the discussion in section 2.2, AIR relies on an accurate approximation to R_{ideal}, and couples this with an accurate F-relaxation scheme to achieve a reduction-based algorithm. In particular, the focus is on reduction through the ideal restriction operator. In [58, Lemma 1], it is shown that for $R \neq R_{ideal}$, care must also be taken when building P to ensure a stable coarse-grid correction. Theory developed here indicates that when R approximates R_{ideal}, building interpolation to accurately capture right singular vectors associated with small singular values is sufficient for a stable coarse-grid correction, as well as two-grid convergence in the ℓ^2- and A^*A-norms. How accurately the derived conditions require P to interpolate modes depends on how accurately $R \approx R_{ideal}$.

This paper can be seen as a companion paper to the ℓAIR method developed in [58]. The theory developed here is more general than that in [58], and provides rigorous explanation as to why better interpolation methods are needed when considering advection-diffusion-reaction [58] compared with pure advection-reaction (here and in [58]). Conversely, the method developed here is less general than ℓAIR, effective primarily on the purely advective or nearly advective case, but as a result, also has a significantly cheaper setup cost. The basic idea is that ℓAIR approximates ideal restriction by solving many small, dense, linear systems, which can be moderately expensive as the problem size grows. nAIR recognizes that, for advective-type discretizations, a similar approximation can be obtained by a few sparse matrix products. A detailed algorithmic comparison of nAIR and ℓAIR can be found in [58].

Finally, theory on convergence of nonsymmetric AMG in the $\sqrt{A^*A}$-norm was developed simultaneously with this work in [40]. There, approximation properties are assumed on R and P. Here, we replace one approximation property assumption on either R or P with a measure of distance of R or P from ideal restriction or interpolation, respectively, and consider convergence in the ℓ^2- and A^*A-norms. If this distance cannot be made small, then one should revert to a framework as in [40], focusing on more traditional approximation properties for P and R.

3. Convergence of reduction-based AMG.

3.1. Framework.

Let A, P, and R take the form introduced in (4). Error propagation of coarse-grid correction is given by $I - \Pi$, where $\Pi = P(RAP)^{-1}RA$ is a (generally nonorthogonal in any known inner product) projection onto the range of P (section 2). The motivation for AIR as an AMG algorithm is straightforward. Recall that ideal restriction gives an exact approximation at C-points, independent of interpolation. Following this with a direct solve on F-points gives an exact two-level method (Lemma 1). Although we do not expect ideal restriction in practice, here we assume that an accurate approximation to R_{ideal} leads to an accurate solution at C-points, which we follow with F-relaxation to distribute this accuracy to F-points.

Let Δ_F be an approximation to A_{ff}^{-1}. First, measures of the accuracy of F-relaxation as well as the difference between ideal interpolation and restriction and the interpolation and restriction used in practice are defined, respectively, as

$$\begin{aligned}\delta_F &= I - \Delta_F A_{ff},\\ \delta_P &= A_{ff}W + A_{fc},\\ \delta_R &= ZA_{ff} + A_{cf}.\end{aligned}$$

Here, δ_R and δ_P are measured relative to $\|A\|$. Throughout the paper it will be assumed that A has been scaled so that $\|A\| \sim O(1)$. However, $\|A\|$ is explicitly carried through the derivation of bounds and proofs of convergence for completeness.

The error-propagation matrix associated with F-point relaxation is given by

$$\mathcal{E}_F = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} \Delta_F & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix} = \begin{bmatrix} \delta_F & -\Delta_F A_{fc} \\ \mathbf{0} & I \end{bmatrix}.$$

The product of these two error matrices, $\mathcal{E} := \mathcal{E}_F(I - \Pi)$, can be put into a very convenient form [58],

$$\begin{aligned}\mathcal{E} &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \left(\begin{bmatrix} \Delta_F & 0 \\ 0 & 0 \end{bmatrix}\right.\\ &\qquad \left. + \begin{bmatrix} I - \Delta_F A_{ff} & -\Delta_F A_{fc} \\ 0 & I \end{bmatrix}\begin{bmatrix} W \\ I \end{bmatrix}\mathcal{K}^{-1}\begin{bmatrix} Z & I \end{bmatrix}\right)\begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix}\\ &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} I & \widehat{W} \\ 0 & I \end{bmatrix}\begin{bmatrix} \Delta_F & 0 \\ 0 & \mathcal{K}^{-1} \end{bmatrix}\begin{bmatrix} I & 0 \\ Z & I \end{bmatrix}\begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix},\end{aligned}$$

where

$$\widehat{W} = (I - \Delta_F A_{ff})W - \Delta_F A_{fc} = \delta_F W - \Delta_F A_{fc}. \tag{7}$$

If $\Delta_F = A_{ff}^{-1}$, then $\widehat{W}$ becomes ideal interpolation. The better Δ_F approximates A_{ff}^{-1}, the closer $\widehat{W}$ is to ideal interpolation. Here, $\widehat{W}$ is referred to as the "effective interpolation" of this method [58].

Next, note that $\mathcal{E}$ has the form

$$\mathcal{E} = I - M^{-1}A = M^{-1}(M - A), \tag{8}$$

where

$$\begin{aligned}M &= \begin{bmatrix} I & 0 \\ -Z & I \end{bmatrix}\begin{bmatrix} \Delta_F^{-1} & 0 \\ 0 & \mathcal{K} \end{bmatrix}\begin{bmatrix} I & -\widehat{W} \\ 0 & I \end{bmatrix}\\ &= \begin{bmatrix} \Delta_F^{-1} & A_{fc} - (\Delta_F^{-1} - A_{ff})W \\ -Z\Delta_F^{-1} & \mathcal{K} + Z\Delta_F^{-1}\widehat{W} \end{bmatrix}. \qquad (9)\end{aligned}$$

A little extra work using (5) yields

$$\begin{aligned}\mathcal{K} + Z\Delta_F^{-1}\widehat{W} &= A_{cc} + A_{cf}W + ZA_{fc} + ZA_{ff}W + Z\Delta_F^{-1}(I - \Delta_F A_{ff})W - ZA_{fc}\\ &= A_{cc} + A_{cf}W + ZA_{ff}W + Z(\Delta_F^{-1} - A_{ff})W\\ &= A_{cc} + A_{cf}W + Z\Delta_F^{-1}W.\end{aligned}$$

Using (9),

$$M - A = \begin{bmatrix} \Delta_F^{-1} - A_{ff} & -(\Delta_F^{-1} - A_{ff})W \\ -(Z\Delta_F^{-1} + A_{cf}) & (Z\Delta_F^{-1} + A_{cf})W \end{bmatrix} \tag{10}$$

$$= \begin{bmatrix} \Delta_F^{-1} - A_{ff} \\ -(Z\Delta_F^{-1} + A_{cf}) \end{bmatrix} \begin{bmatrix} I & -W \end{bmatrix}. \tag{11}$$

Similarly, M^{-1} can be expanded as

$$\begin{aligned} M^{-1} &= \begin{bmatrix} I & \widehat{W} \\ 0 & I \end{bmatrix} \begin{bmatrix} \Delta_F & 0 \\ 0 & \mathcal{K}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ Z & I \end{bmatrix} \\ &= \begin{bmatrix} \Delta_F + \widehat{W}\mathcal{K}^{-1}Z & \widehat{W}\mathcal{K}^{-1} \\ \mathcal{K}^{-1}Z & \mathcal{K}^{-1} \end{bmatrix}. \end{aligned} \tag{12}$$

As mentioned in (8), the error-propagation matrix is given by $\mathcal{E} = M^{-1}(M - A)$. The residual-propagation matrix is similar, given by $\mathcal{R} = A\mathcal{E}A^{-1} = (M - A)M^{-1}$. Each of these can now be assembled to a convenient outer-product form.

For error propagation, combining (11) with (12) gives

$$\begin{aligned} \mathcal{E} &= \begin{bmatrix} \Delta_F + \widehat{W}\mathcal{K}^{-1}Z & \widehat{W}\mathcal{K}^{-1} \\ \mathcal{K}^{-1}Z & \mathcal{K}^{-1} \end{bmatrix} \begin{bmatrix} \Delta_F^{-1} - A_{ff} \\ -(Z\Delta_F^{-1} + A_{cf}) \end{bmatrix} \begin{bmatrix} I & -W \end{bmatrix} \\ &= \begin{bmatrix} (I - \Delta_F A_{ff}) - \widehat{W}\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) \\ -\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) \end{bmatrix} \begin{bmatrix} I & -W \end{bmatrix} \\ &= \begin{bmatrix} \delta_F - \widehat{W}\mathcal{K}^{-1}\delta_R \\ -\mathcal{K}^{-1}\delta_R \end{bmatrix} \begin{bmatrix} I & -W \end{bmatrix}. \end{aligned} \tag{13}$$

Likewise, using (7), (11), and (12), the residual-propagation matrix is given by

$$\mathcal{R} = \begin{bmatrix} \Delta_F^{-1} - A_{ff} \\ -(Z\Delta_F^{-1} + A_{cf}) \end{bmatrix} \begin{bmatrix} \Delta_F(I - \delta_P\mathcal{K}^{-1}Z) & -\Delta_F\delta_P\mathcal{K}^{-1} \end{bmatrix}. \tag{14}$$

The outer-product formulation provides a natural representation of powers of $\mathcal{E}^k$ and $\mathcal{R}^k$, where

$$\mathcal{E}^k = \begin{bmatrix} \delta_F - \widehat{W}\mathcal{K}^{-1}\delta_R \\ -\mathcal{K}^{-1}\delta_R \end{bmatrix} G^{k-1} \begin{bmatrix} I & -W \end{bmatrix}, \tag{15}$$

$$\mathcal{R}^k = \begin{bmatrix} \Delta_F^{-1} - A_{ff} \\ -(Z\Delta_F^{-1} + A_{cf}) \end{bmatrix} G^{k-1} \begin{bmatrix} \Delta_F(I - \delta_P\mathcal{K}^{-1}Z) & -\Delta_F\delta_P\mathcal{K}^{-1} \end{bmatrix}. \tag{16}$$

In particular, it is easily verified from (7), (13), and (14) that

$$\begin{aligned} G &= \begin{bmatrix} I & -W \end{bmatrix} \begin{bmatrix} \delta_F - \widehat{W}\mathcal{K}^{-1}\delta_R \\ -\mathcal{K}^{-1}\delta_R \end{bmatrix} \\ &= \delta_F + \Delta_F\delta_P\mathcal{K}^{-1}\delta_R \qquad (17) \\ &= \delta_F + (I - \delta_F)A_{ff}^{-1}\delta_P\mathcal{K}^{-1}\delta_R \qquad (18) \end{aligned}$$

is identical for $\mathcal{E}^k$ and $\mathcal{R}^k$.

This is a fundamental observation for proving convergence-in-norm of reduction-based AMG. In particular, it is often the case that $\|\mathcal{E}\|$ and/or $\|\mathcal{R}\|$ are greater than one. However, (15) and (16) show that raising error- and residual-propagation to powers is equivalent to considering powers of a different matrix, G (18). Thus, bounding $\|G\|$ can lead to a convergent method, which is summarized in the following lemma.

Lemma 3. *Let W, Z, and Δ_F be chosen such that*

$$\|G\| = \|\delta_F + \Delta_F \delta_P \mathcal{K}^{-1}\delta_R\| = \rho < 1.0.$$

Then, the iteration will converge with bounds

$$\|\mathbf{e}_k\| \leq \rho^{k-1} \left\| \begin{bmatrix} \delta_F - \widehat{W}\mathcal{K}^{-1}\delta_R \\ -\mathcal{K}^{-1}\delta_R \end{bmatrix} \right\| \|\begin{bmatrix} I & -W \end{bmatrix}\| \, \|\mathbf{e}_0\|,$$

$$\|\mathbf{r}_k\| \leq \rho^{k-1} \left\| \begin{bmatrix} \Delta_F^{-1} - A_{ff} \\ -(Z\Delta_F^{-1} + A_{cf}) \end{bmatrix} \right\| \|\begin{bmatrix} \Delta_F(I - \delta_P \mathcal{K}^{-1} Z) & -\Delta_F \delta_P \mathcal{K}^{-1} \end{bmatrix}\| \, \|\mathbf{r}_0\|.$$

Proof. The proof follows from (15), (16), (17), and the discussion above. □

Note that the bound on $\|r_k\|$ is independent of $\|A\|$, but the separate terms are not. This can easily be adjusted by scaling the second and third terms by $1/\|A\|$ and $\|A\|$, respectively.

In section 4.1, we show that it is often possible in practice to construct Z such that $\|\delta_R\|$ is quite small relative to $\|A\|$. Recall that, for hyperbolic problems, $\|\mathcal{K}^{-1}\| = O(1/h)$. On a relatively coarse grid, it is possible that $\|\delta_R\| \ll O(h)$ and, consequently, $\|G\| < 1.0$, regardless of W. In fact, $W = \mathbf{0}$ is a reasonable choice in that context because that choice reduces the complexity of $\mathcal{K}$, making the algorithm more efficient. However, in general, a better W may be necessary for convergence. The following section develops conditions for which $\|G\| < 1$.

Remark 4 (pre-F-relaxation). Because ideal restriction gives an exact coarse-grid correction at C-points, thus far we have considered post-F-relaxation to distribute this accuracy to F-points. If, instead, P is chosen to approximate P_{ideal}, a pre-F-relaxation scheme may be more appropriate (see Corollary 2). It is easy to show that pre-F-relaxation enjoys the same asymptotic behavior as post-F-relaxation.

3.2. Two-grid convergence.

In this section, conditions are derived to bound $\|G\| \leq \rho < 1$. The focus of this work is on problems for which $\|\delta_F\|$, and $\|\delta_R\|$ or $\|\delta_P\|$, can be made small relative to $\|A\|$. However, for a given family of discretizations, $\|\delta_R\|$ and $\|\delta_P\|$ are typically fixed, independent of h; that is, the accuracy of approximation to ideal operators does not improve as $h \to 0$. To bound $\|G\| < 1$ as $h \to 0$, additional measures must be taken to account for the term $\delta_P \mathcal{K}^{-1}\delta_R$ in (18), because $\|\mathcal{K}^{-1}\| \sim O(1/h)$. To do so, we consider a classical multigrid “weak approximation property” for P and R.

Definition 5 (WAP on P with respect to SPD $\mathcal{A}$). *An interpolation operator, P, satisfies the weak approximation property (WAP) with respect to SPD matrix $\mathcal{A}$, with constant C_P if, for any $\mathbf{v}$ on the fine grid, there exists a $\mathbf{w}_c$ on the coarse grid such that*

$$\|\mathbf{v} - P\mathbf{w}_c\|^2 \leq \frac{C_P}{\|\mathcal{A}\|}\langle \mathcal{A}\mathbf{v}, \mathbf{v}\rangle.$$

Recall that the Schur complement of A is $\mathcal{K}_A = A_{cc} - A_{cf}A_{ff}^{-1}A_{fc}$. Comparing the coarse-grid operator (5) with the Schur complement yields

$$\begin{aligned} \mathcal{K} - \mathcal{K}_A &= (ZA_{ff} + A_{cf})A_{ff}^{-1}A_{fc} + (ZA_{ff} + A_{cf})W \\ &= (ZA_{ff} + A_{cf})A_{ff}^{-1}(A_{ff}W + A_{fc}) \\ &= \delta_R A_{ff}^{-1}\delta_P. \end{aligned} \tag{19}$$

Now, assume that P satisfies the WAP with respect to $\mathcal{A} = A^*A$, with constant C_P. Then, for every $\mathbf{v} = (\mathbf{v}_f^T, \mathbf{v}_c^T)^T$,

$$\inf_{\mathbf{w}_c} \left\| \begin{pmatrix} \mathbf{v}_f \\ \mathbf{v}_c \end{pmatrix} - \begin{pmatrix} W \\ I \end{pmatrix} \mathbf{w}_c \right\|^2 \leq \frac{C_P}{\|A^*A\|}\|A\mathbf{v}\|^2. \tag{20}$$

Let $\hat{\mathbf{w}}_c$ satisfy the infimum above. Then,

$$\begin{aligned} \|\mathbf{v}_f - W\mathbf{v}_c\| &\leq \|\mathbf{v}_f - W\hat{\mathbf{w}}_c\| + \|W(\hat{\mathbf{w}}_c - \mathbf{v}_c)\| \\ &\leq \|\mathbf{v}_f - W\hat{\mathbf{w}}_c\| + \|W\|\|\hat{\mathbf{w}}_c - \mathbf{v}_c\|. \end{aligned}$$

Noting from (20) that $\|\mathbf{v}_f - W\hat{\mathbf{w}}_c\|^2 + \|\mathbf{v}_c - \hat{\mathbf{w}}_c\|^2 \leq \frac{C_P}{\|A^*A\|}\|A\mathbf{v}\|^2$, we can form a constrained maximization problem and bound

$$\|\mathbf{v}_f - W\mathbf{v}_c\| \leq \sqrt{C_P(1 + \|W\|^2)}\frac{\|A\mathbf{v}\|}{\|A\|} := C_W\frac{\|A\mathbf{v}\|}{\|A\|},$$

where $C_W := \sqrt{C_P(1 + \|W\|^2)}$. In particular, let $\mathbf{v}_f = -A_{ff}^{-1}A_{fc}\mathbf{v}_c$. Then,

$$\|A_{ff}^{-1}\delta_P\mathbf{v}_c\| = \|(W + A_{ff}^{-1}A_{fc})\mathbf{v}_c\| \leq C_W\frac{\|A\mathbf{v}\|}{\|A\|} = C_W\frac{\|\mathcal{K}_A\mathbf{v}_c\|}{\|A\|}. \tag{21}$$

Following from (19) and (21), observe that

$$\begin{aligned} \|\mathcal{K}\mathbf{v}_c\| &\geq \|\mathcal{K}_A\mathbf{v}_c\| - \|(\mathcal{K}_A - \mathcal{K})\mathbf{v}_c\| \\ &= \|\mathcal{K}_A\mathbf{v}_c\| - \|\delta_R A_{ff}^{-1}\delta_P\mathbf{v}_c\| \\ &\geq \left(1 - C_W\frac{\|\delta_R\|}{\|A\|}\right)\|\mathcal{K}_A\mathbf{v}_c\|. \end{aligned}$$

Note that, given W with WAP constant C_W, we may choose Z so that $C_W\|\delta_R\| < \|A\|$. Later, $C_W\|\delta_R\|$ will be chosen slightly smaller. Combining, we arrive at

$$\|A_{ff}^{-1}\delta_P\mathcal{K}^{-1}\| = \sup_{\mathbf{v}_c \neq \mathbf{0}} \frac{\|A_{ff}^{-1}\delta_P\mathbf{v}_c\|}{\|\mathcal{K}\mathbf{v}_c\|} \leq \frac{C_W}{\|A\| - C_W\|\delta_R\|}.$$

This result is generalized in the following lemma.

Lemma 6. *Suppose P satisfies the WAP with respect to A^*A, with constant C_P. Then, if Z is chosen such that $C_W\|\delta_R\| < \|A\|$,*

$$\|A_{ff}^{-1}\delta_P\mathcal{K}^{-1}\| \leq \frac{C_W}{\|A\| - C_W\|\delta_R\|}, \qquad \|\delta_P\mathcal{K}^{-1}\| \leq \frac{C_W\|A_{ff}\|}{\|A\| - C_W\|\delta_R\|},$$

where $C_W = C_P\sqrt{1 + \|W\|^2}$. Similarly, suppose R^* satisfies the WAP with respect to AA^*, with constant C_R. Define $C_Z = \sqrt{C_R(1 + \|Z\|^2)}$ and assume that $C_Z\|\delta_P\| < \|A\|$. Then,

$$\|\mathcal{K}^{-1}\delta_R A_{ff}^{-1}\| \leq \frac{C_Z}{\|A\| - C_Z\|\delta_P\|}, \qquad \|\mathcal{K}^{-1}\delta_R\| \leq \frac{C_Z\|A_{ff}\|}{\|A\| - C_Z\|\delta_P\|}.$$

Proof. The results follow from the above discussion and noting that $\|\delta_P\mathcal{K}^{-1}\| \leq \|A_{ff}\|\|A_{ff}^{-1}\delta_P\mathcal{K}^{-1}\|$. Results for R follow a similar derivation as that for P. □

The above discussion is summarized in the following results.

Theorem 7. *Let A, P, and R be defined as above. If W is chosen so that P satisfies the WAP with respect to A^*A, with constant C_P, then Z can be chosen to approximate $-A_{cf}A_{ff}^{-1}$ and Δ_F can be chosen to approximate A_{ff}^{-1} so that*

$$\|G\| \leq \|\delta_F\| + \Big(1 + \|\delta_F\|\Big)\frac{C_W\|\delta_R\|}{\|A\| - C_W\|\delta_R\|} = \rho < 1,$$

where $C_W = C_P\sqrt{1 + \|W\|^2}$.

Proof. The proof follows from Lemma 6 and the bound using (17),

$$\|G\| \leq \|\delta_F\| + (1 + \|\delta_F\|)\|\delta_R\|\|A_{ff}^{-1}\delta_P\mathcal{K}^{-1}\|. \qquad \square$$

Theorem 8. *Let A, P, and R be defined as above. If Z is chosen so that R satisfies the WAP with respect to AA^* and constant C_R, then W can be chosen to approximate $-A_{ff}^{-1}A_{fc}$ and Δ_F can be chosen to approximate A_{ff}^{-1} such that*

$$\|G\| \leq \|\delta_F\| + (1 + \|\delta_F\|)\frac{C_Z\|\delta_P\|}{\|A\| - C_Z\|\delta_P\|} = \rho < 1,$$

where $C_Z = C_R\sqrt{1 + \|Z\|^2}$.

Proof. The proof follows from the discussion above and the proof of Theorem 7. □

Theorems 7 and 8 give insight into the roles of restriction, interpolation, and F-relaxation. F-relaxation can help convergence bounds, but only to a certain extent. For an exact solve on F-points, $\|\delta_F\| = 0$. Then, for example, in Theorem 7, to ensure that $\|G\| \leq \rho < 1$, we must have $C_W\|\delta_R\|/\|A\| \leq \frac{\rho}{\rho+1} < \frac{1}{2}$. This can be accomplished both through a more accurate interpolation with respect to the WAP or a more accurate approximation to ideal restriction. As $\|\delta_F\|$ increases, that is, F-relaxation becomes less effective; interpolation and restriction must improve through reduced C_W and/or $\|\delta_R\|$.

The ℓ^2-convergence of error and residual also follows from Theorems 7 and 8 and Lemma 3. Although it is possible that $\|\mathcal{E}\|, \|\mathcal{R}\| > 1$, if the hypotheses are satisfied, there exists k_1 and k_2 such that $\|\mathcal{E}^{k_1}\|, \|\mathcal{R}^{k_2}\| < 1$. Iterations before these values of k are reached may appear to be diverging, but they will eventually converge with asymptotic factor ρ. How long it takes to reach asymptotic convergence depends on the other matrix blocks in $\mathcal{E}, \mathcal{R}$. This has been observed in practice.

Consider $\|\mathcal{E}\|$ and $\|\mathcal{R}\|$ with respect to the size of the mesh. From (13), it is clear that choosing $\mathbf{e} = (\mathbf{0}, \mathbf{e}_c^T)^T$ yields $\|\mathcal{E}\| \geq \|\mathcal{K}^{-1}\delta_R\|$. In the case when A is a discrete approximation of a PDE, $\|\mathcal{K}^{-1}\|$ may grow with n, the size of the system, whereas $\|\delta_R\|$ may be fixed. Although $\|G\| = \rho$ is independent of n, without additional approximation properties on Z, the norm of $\mathcal{E}$ may not be bounded independent of n. This would lead to a method for which it takes more iterations to reach a given accuracy as the problem size increases. Building on Lemma 6, conditions for residual and error propagation bounded independent of n are given in the following corollary.

Corollary 9 (bounded residual and error). *Assume that P satisfies the WAP with respect to A^*A, with constant C_P, Theorem 7 holds, and the condition numbers* *of A_{ff} and Δ_F are independent of problem size. Then, for $k \geq 0$, $\|\mathcal{R}^k\|$ is bounded, independent of problem size, and converges with asymptotic rate $\leq \rho$.*

If, in addition, R^* satisfies a WAP with respect to AA^*, then for $k \geq 0$, $\|\mathcal{E}^k\|$ is bounded, independent of problem size, and converges with asymptotic rate $\leq \rho$.

Proof. Consider the terms in (16). Under the assumption that P satisfies a WAP with respect to A^*A, it was shown in Lemma 6 that $\|\delta_P\mathcal{K}^{-1}\|$ is bounded independent of n. All other terms in the equation are bounded independent of n. Likewise, consider the terms in (15). From Lemma 6, if R^* satisfies a WAP with respect to AA^*, then $\|\mathcal{K}^{-1}\delta_R\|$ is bounded independent of n. □

Remark 10 (C-point relaxation). In a two-level setting, adding relaxation over C-points as part of the pre- or postrelaxation scheme offers little to no improvement of convergence. Suppose $Z = -A_{cf}\Delta_R$, where $\Delta_F = \Delta_R$, that is, the same approximation is used for F-relaxation as for approximating R_{ideal}. Now, consider following the F-point relaxation with a C-point relaxation. Similar to F-point relaxation, the error-propagation operator associated with C-point relaxation is given by

$$\mathcal{E}_C = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \Delta_C \end{bmatrix} \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix},$$

where Δ_C is an approximation to A_{cc}^{-1}. Noting that the C-point rows of $(M - A)$ are zero when $Z = -A_{cf}\Delta_F$ (see (10)), multiplying by $\mathcal{E}$ yields

$$\begin{aligned} \mathcal{E}_C\mathcal{E} &= I - \left(M^{-1} + \begin{bmatrix} 0 & 0 \\ 0 & \Delta_C \end{bmatrix} (M - A)M^{-1}\right) A \\ &= I - M^{-1}A = \mathcal{E}_E. \end{aligned}$$

This demonstrates that, in the context of a two-grid method, with a reasonable choice of Z, C-point relaxation does not improve the solution. In the multilevel setting, C-point relaxation can offer some improvement in convergence, but remains much less important than F-relaxation.

4. The triangular case.

Building on the previous section, this section develops an accurate approximation to R_{ideal} for matrices with block-triangular or near-triangular structure. Numerical examples demonstrate the accuracy of nAIR in the context of convergence constants derived in section 3. Although reduction can be achieved with ideal restriction or interpolation, focusing on ideal restriction allows for coupling the nAIR method with established interpolation methods. One result here is that error and residual propagation of nAIR are nilpotent in the case of block-triangular matrices, which compensates for inaccurate interpolation near boundaries.

As motivation, consider a block-discontinuous discretization of a steady state advection or advection-reaction equation in two dimensions,

$$\mathbf{b} \cdot \nabla u + c(x, y)u = f,$$

for arbitrary velocity field $\mathbf{b}(x, y)$ (without cycles, or else the problem is not well posed), forcing function $\mathbf{f}$, reaction field $c(x, y)$, and some inflow and outflow boundary conditions (for example, see section 6). Suppose a uniform square mesh is used in two dimensions. Then, for many discretizations, such as discontinuous Galerkin, among others, DOFs in each element of the mesh depend on exactly two other elements in the mesh, specifically the two elements upwind with respect to the direction of flow, and each element in the mesh corresponds to a nonoverlapping block in the matrix. In

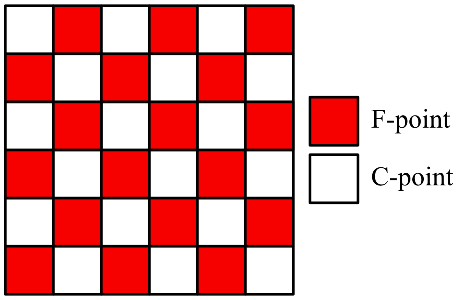


Fig. 1. *Elementwise "block" white-red (or white-gray) coarsening of uniform structured grid in two dimensions.*

the multigrid context, consider a block red-black coarsening scheme, where C-points and F-points represent entire finite elements, as shown in Figure 1.

For linear finite elements, the DOFs corresponding to a block discretization are often located at the vertices of the mesh, with one DOF contained in each element touching a given vertex. For higher-order discretizations, there are more DOFs, but the underlying principle remains the same: all DOFs in a given finite element only depend on DOFs within that element or DOFs in either of the two upwind elements. Looking at Figure 1, note that for any velocity field $\mathbf{b}(x, y)$, all F-point blocks depend only on C-point blocks, and C-point blocks depend only on F-point blocks. If A is scaled to have unit diagonal, then it follows that the submatrix $A_{ff} = I$, and thus, ideal restriction and exact F-relaxation are trivial to compute in practice.

Of course, this example holds for a block diffusion discretization as well (that is, $A_{ff} = I$ there as well). However, for advection, the coarse-grid operator $RAP = A_{cc} - A_{cf}A_{ff}^{-1}A_{fc} = I - A_{cf}A_{fc}$ maintains a structure similar to the fine grid. Note that for any C-point, the corresponding row of $RAP = I - A_{cf}A_{fc}$ is nonzero in C-point blocks that can be reached through a C-F-C path in the graph. Generally for an advection problem, (i) each C-point has about three coarse-grid connections, (ii) all connections are upwind (and, thus, strictly lower triangular in the matrix), and (iii) at least one of these connections is essentially cross-stream, making it a weak connection. In this case, coarsening similar to Figure 1 based on strong connections leads to an $A_{ff} \neq I$, but for which A_{ff}^{-1} is well conditioned and easily approximated with a sparse matrix. In contrast, each coarse-grid point in a diffusion discretization is connected to eight other points, making A_{ff}^{-1} and R_{ideal} difficult to approximate effectively in a sparse manner, and multilevel reduction much more difficult.

On unstructured meshes, nonquadrilateral elements, higher dimensions, or coarser grids in an AMG hierarchy, it is typically not the case that $A_{ff} = I$. Nevertheless, similar principles suggest that A_{ff}^{-1} can be approximated efficiently, which is confirmed directly in section 4.2 and implicitly in numerical results of section 6.

4.1. Neumann approximate ideal restriction.

For general matrices, a naive and often ineffective approach to approximate A_{ff}^{-1} is to use a truncated Neumann expansion. However, in the case of block-triangular matrices, particularly those resulting from the discretization of differential operators, a truncated Neumann inverse expansion can provide a remarkably accurate approximation. For ease of notation, assume that A has been scaled to have unit diagonal, and suppose we have determined a CF-splitting (or block CF-splitting if A is block lower triangular). Then, let

$A_{ff} = I - L_{ff}$, where L_{ff} is the strictly lower triangular part of A_{ff}. Because L_{ff} is nilpotent, A_{ff}^{-1} can be written as a finite Neumann expansion:[2]

$$A_{ff}^{-1} = (I - L_{ff})^{-1} = \sum_{i=0}^{d_f+1} L_{ff}^i. \tag{22}$$

To approximate A_{ff}^{-1}, we consider an order-k approximation given by truncating (22): $\Delta^{(k)} := \sum_{i=0}^{k} L_{ff}^i$, for some $0 \leq k \leq d_f$. Define a restriction operator based on a Neumann approximate ideal restriction (nAIR):

$$R := \begin{bmatrix} -A_{cf}\Delta^{(k)} & I \end{bmatrix}.$$

Error in $\Delta^{(k)}$ as an approximate inverse can be measured as $I - \Delta^{(k)}A_{ff} = L_{ff}^{k+1}$, which gives a measure of how accurately we approximate R_{ideal}.

Note that the error relation $I - \Delta^{(k)}A_{ff} = L_{ff}^{k+1}$ does not require A_{ff} to be lower triangular. However, triangular structure is fundamental to L_{ff}^k being small as k increases, particularly when considering the discretization of differential operators. Consider L_{ff} as the adjacency matrix of a directed acyclic graph. Then $(L_{ff}^k)_{ij}$ gives the sum of weighted walks of length k from vertex i to vertex j (weight given by the *product* of the walk's edges). Thus, we are interested in the number of F-F connections and size of the weights. For the discretization of differential operators, off-diagonal elements are typically small relative to the diagonal, and an AMG CF-splitting is chosen to eliminate strong F-F connections. In the case of triangular matrices, such as an upwind discretization of advection, regardless of the problem dimension, there only exist walks from node i to nodes j *downstream* of i, in the direction along the characteristic. This means that the sparsity pattern of L_{ff}^k only reaches out to neighbors in effectively one direction. Thus, as k increases, the number of neighbors within distance k should not increase significantly, while the product of edges should decay rapidly.

Remark 11 (nilpotent error propagation). It is straightforward to show that if A is lower triangular in some ordering and Δ_R, Δ_P, and Δ_F are truncated Neumann approximations to A_{ff}^{-1}, then two-grid error propagation based on Jacobi F-relaxation is strictly lower triangular. Moreover, multilevel error propagation, with coarse-grid correction based on nAIR and Jacobi F-relaxation, is also strictly lower triangular and, thus, nilpotent.

Although, the degree of nilpotency is not sufficiently small to be considered practical, nilpotency of error propagation is directly impactful near the boundaries of the domain, where DOFs actually fall off the nilpotent cliff in $O(1)$ iterations. AMG often struggles with interpolation near domain boundaries, and the nilpotent behavior of nAIR eliminates this problem. This benefit has been observed in practice.

4.2. Evaluating constants.

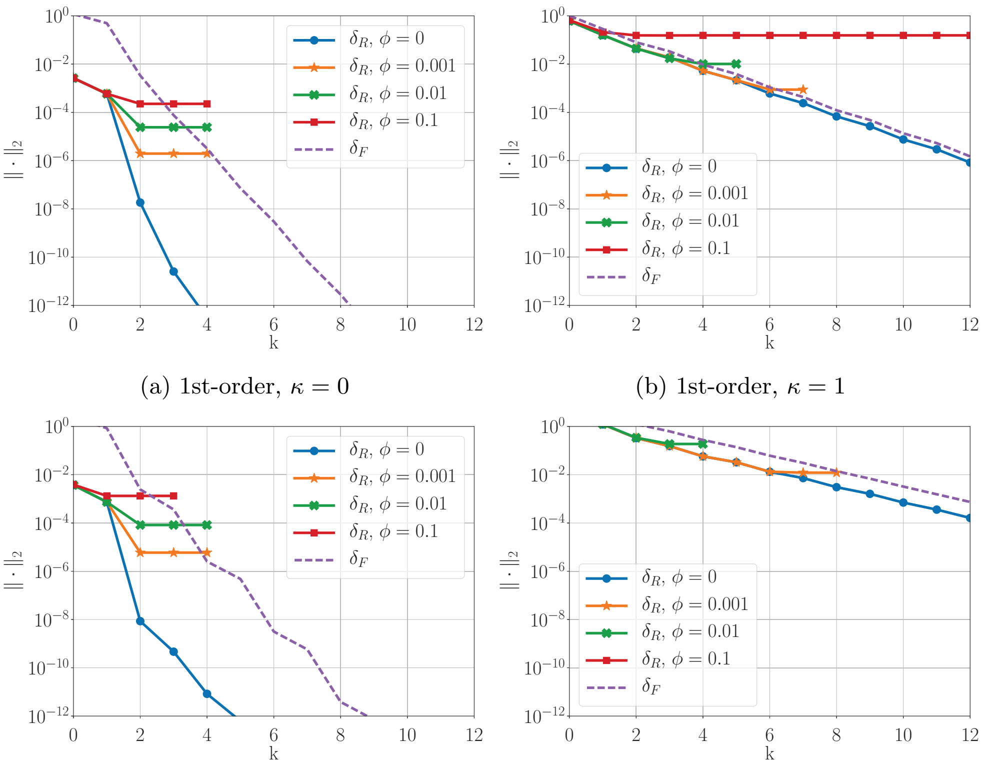
This section considers nAIR applied to discontinuous Galerkin discretizations of the steady state transport equation and an advection-diffusion-reaction equation, with diffusion coefficient κ. Both discretizations are on

[2] Because there are no cycles in the graph of L_{ff}, the degree of nilpotency is given by the maximum graph distance between any two F-points, say d_f. In the case of an acyclic graph, this is equivalent to the *longest path problem*. For discretizations of differential operators that result in an acyclic graph, it is generally the case that $d_f \ll n_f$.

unstructured meshes with a moving (nonconstant) velocity field and material discontinuities of 10^8 between different subdomains. Further details on the transport discretization can be found in section 6. The advection-diffusion-reaction equation follows an analogous derivation, and is discussed in [58]. Recall that, when approximating ideal restriction (as opposed to ideal interpolation), the theoretical constants of interest are

$$\delta_F = I - \Delta_F A_{ff}, \qquad \delta_R = ZA_{ff} + A_{cf},$$

as well as the constant for which P satisfies a WAP (see Theorem 7). The purpose of this section is to demonstrate that, on an interesting and difficult model problem, nAIR leads to $\|\delta_F\| \ll 1, \|\delta_R\| \ll \|A\|$, where A has been scaled so that $\|A\| \sim O(1)$. Following from Theorem 7, we then show that strong two-grid convergence is provable under minimal assumptions on P for the transport equation. Results also show why the reduction approach is successful for advection discretizations, but less effective for diffusive problems.



(a) 1st-order, $\kappa = 0$

(b) 1st-order, $\kappa = 1$

(c) 4th-order, $\kappa = 0$

(d) 4th-order, $\kappa = 1$

FIG. 2. *Using upwind discontinuous Galerkin (DG), $\|\delta_R\|$ and $\|\delta_F\|$ are displayed as a function of k, corresponding to degree of Neumann expansion for nAIR and number of (block) smoothing iterations for Jacobi F-relaxation. The top row corresponds to linear finite elements and the bottom to 4th-order finite elements, while the left column corresponds to steady state transport (no diffusion, $\kappa = 0$) and the right columns corresponds to a diffusion-dominated advection-diffusion-reaction equation, $\kappa = 1$. δ_R is a function of parameter $\phi < 1$, where a Neumann expansion is only performed on strong connections (for row i, a_{ij} such that $|a_{ij}| \geq \phi \sup_j |a_{ij}|$). In all cases, the discretization has $\approx 400,000$ DOFs, about half of which are F-points.*

Figure 2 plots $\|\delta_F\|$ and $\|\delta_R\|$ as a function of the number of iterations and degree of Neumann expansion, respectively. In practice, it is usually not a good idea to form transfer operators using neighbors further apart than distance two, unless coarsening aggressively, due to the rapid increase in the number of nonzeros in coarse-grid operators. Similarly, transfer operators are typically only formed based on strong connections in the matrix, again to limit coarse-grid fill-in, particularly in the multilevel setting. Here, strong connections for each row are defined as entries larger than ϕ times the largest row element (positive or negative).

Looking at the left column of Figure 2, we see that for both 1st- and 4th-order discretizations of the steady state transport problem, nAIR based on distance-one or -two strong neighbors is able to achieve $\|\delta_R\|$ between $10^{-5}-10^{-3}$. Furthermore, three iterations of F-relaxation leads to $\|\delta_F\| \approx 10^{-4}$, and four iterations improves this to $\|\delta_F\| \approx 10^{-6}$. In the diffusion-dominated case, $\kappa = 1$ (right column), results are not as good. For linear elements, distance-two nAIR achieves at best $\|\delta_R\| \approx 0.1$, and four iterations of F-relaxation only yields $\|\delta_F\| \approx 0.01$, while the 4th-order discretization is worse.

Table 1 plots two-grid convergence bounds from Theorem 7 as a function of constants C_W and $\|\delta_R\|$, for a fixed $\|\delta_F\|$. Note that due to small $\|\delta_R\|$ and $\|\delta_F\|$, only mild assumptions must be made on interpolation for rapid two-grid convergence. Although constants are likely to increase somewhat in the multilevel setting, the initial discussion in section 4 suggests that good approximations to A_{ff}^{-1} and R_{ideal} can still be obtained on coarser levels in the hierarchy.

Table 1

Theoretical two-grid convergence bounds as a function of $\|\delta_R\|$ and C_W, assuming $\|\delta_F\| = 10^{-4}$ and A has been scaled such that $\|A\| \sim O(1)$.

$\|\delta_R\|$ / C_W	1	5	10	25	50	100	250	500
10^{-5}	0.0001	0.0002	0.0002	0.0004	0.0006	0.001	0.002	0.005
10^{-4}	0.0002	0.0006	0.0011	0.0026	0.0051	0.010	0.026	0.053
10^{-3}	0.0011	0.0051	0.0102	0.0257	0.0527	0.111	0.334	–
0.01	0.0102	0.0527	0.1112	0.3335	–	–	–	–
0.1	0.1112	–	–	–	–	–	–	–

The constant C_P, where $C_W := \sqrt{C_P(1 + \|W\|^2)}$, is evaluated in [40] for a hyperbolic steady state transport equation, discretized with SUPG and DG finite element discretizations (for details, see section 6 here or [40]). There, modified classical AMG interpolation [19] is shown to have constant C_P of 157 and 204 for SUPG and DG, respectively. Typically, $\|W\| \gtrapprox 1$. Say $\|W\| = 2$. Then $C_W \approx 28$ and 32. Applying the same tests to the one-point interpolation used here yields constants C_P on the order of 300 and 360, and $C_W \approx 39$ and 42. In practice, we find that the sparser structure of one-point interpolation is advantageous in the multilevel setting, and we see equivalent convergence with the two methods, despite the larger constant C_W using one-point interpolation.

Interestingly, in [40] an algorithm analogous to AIR for interpolation is numerically shown to have constants $C_P \approx 10 - 20$ and $C_W \approx 7$ and 10. Furthermore, tests suggest the constants may be independent of problem size, a property which may not be the case for one-point and classical interpolation. Note that those tests did not use an efficient algorithm to build the interpolation, and extending AIR-like ideas to interpolation for hyperbolic-type problems is ongoing work. However, such numbers indicate interpolation operators that are better than existing methods can be constructed for hyperbolic-type problems, and, with such interpolation, two-grid convergence could be obtained with fairly weak approximations of R_{ideal}.

5. Algorithm.

The main components of nAIR follow that of a standard Petrov–Galerkin multigrid scheme with no prerelaxation. Section 5.1 introduces details on the AMG components of nAIR, including parameters and routines for strength-of-connection, CF-splitting, interpolation, restriction, and relaxation. Other details, including support of block structure in matrices and a filtering procedure to reduce complexity, are discussed in section 5.2.

5.1. AMG components.

Effective F-relaxation and an accurate approximation to ideal restriction both require A_{ff} to be relatively well conditioned. This is consistent with motivation for a classical AMG strength-of-connection and CF-splitting [10, 54], which are used here. Jacobi F-relaxation is used with one more iteration than the degree of the Neumann approximation of R_{ideal}. Restriction is built using a degree-one Neumann expansion (22) applied to strong connections in A_{ff}, with connection drop-tolerance $\phi = 0.025$. These parameters are motivated through the comparison of nAIR with R_{ideal} in section 4.1.

From Theorem 7, if we approximate R_{ideal}, then P should be built targeting approximation properties in a classical AMG sense. It turns out that classical AMG interpolation formulae [10, 54] do not satisfy approximation properties on hyperbolic transport discretizations [40]. Fortunately, R is a very accurate approximation to R_{ideal} for problems tested here, making interpolation less important. We propose a "one-point interpolation" scheme where each F-point is interpolated by value from its strongest C-connection. One-point interpolation resembles a degree-zero Neumann expansion, but P is sparser, having exactly one nonzero per row, and each nonzero is set to one as opposed to the value of A_{fc}.[3] This ensures that the constant is in the range of interpolation, with minimal nonzero requirements of P. In practice, one-point interpolation performs best compared to many different interpolation methods tested in terms of total work and time to solution. In fact, ℓAIR was shown to have good approximation properties for scalar hyperbolic problems in [40], and using these ideas to develop improved interpolation methods for hyperbolic problems is ongoing work.

5.2. Blocks, filtering, and parallelization.

Some PDE discretizations lead to matrix equations with a natural block structure. The two most common examples are as follows: (i) systems of PDEs, where a block in the matrix corresponds to multiple variables discretized on a single spatial node, or (ii) block discontinuous discretizations, such as DG, where each finite element forms a block in the matrix. In either case, a block lower triangular matrix can be transformed into lower triangular by scaling the system by the block-diagonal inverse, or nAIR can be performed in the block setting, coarsening and forming transfer operators by block. For most results, we scale by the block-diagonal inverse because the two approaches have shown comparable convergence factors, and forming nAIR in the setup phase is cheaper and simpler in the scalar (nonblock) matrix case. However, section 6.3 shows results for block nAIR as well, where coarsening, restriction, interpolation, and relaxation are all done in a block fashion.

One way to further reduce complexity in an AMG solver is truncating or lumping

[3] One-point interpolation also resembles an unsmoothed aggregation interpolation operator, but here some "aggregates" may have many points and others only one. In aggregation-based AMG, aggregates are typically chosen to be comparable in size, and there are never aggregates of size one that have strong connections in the matrix.

operators. The idea is simple: remove entries from a matrix in the hierarchy, A_ℓ, that are smaller than some threshold, typically with respect to the diagonal element of the given row. Such methods have been used in AMG for symmetric problems with diffusive components [9, 25, 59]. Heuristically, eliminating small entries is even more appropriate in the hyperbolic setting, because the solution at any given point only depends on the solution at other points upwind along the characteristic. In the discrete setting, small entries that arise in matrix operations are often not aligned with the characteristic and are more of a numerical effect, suggesting that some can be eliminated without degrading convergence. Numerical results confirm this; in particular, removing entries in the case of SPD matrices is a delicate process [9, 25, 59], but section 6 shows that entries can be removed from discretizations of steady state transport aggressively without a degradation in convergence. For some drop-tolerance φ, elements $\{a_{ij} \mid j \neq i, |a_{ij}| \leq \varphi|a_{ii}|\}$ are eliminated (that is, set to zero) for each row i of matrix A_ℓ. A drop tolerance of $\varphi = 10^{-3}$ has proved to be effective for many problems tested and is used for all results presented here.

Finally, nAIR applied to triangular systems is intended for highly parallel environments, where traditional triangular solves are not easily parallelized. This work focuses on algorithmic details and theory, and does not develop parallel performance models or present parallel scaling results. However, it is well known that AMG scales in parallel to hundreds of thousands of cores [6], with a communication cost of $O(\log P)$, for P processors [24, 32]. The algorithm developed here takes on the form of a traditional AMG method, with the additional cost of building and storing a restriction operator, which can be performed efficiently in parallel.

6. Numerical results.

In this section, we apply nAIR to discontinuous discretizations of the steady state transport equation. For problems in which the steady state is well posed (no cycles in flow), the steady state case is equivalent to the time dependent problem with an infinite time step. In this context, successful results on steady state flow also indicate that nAIR is applicable in the time-dependent regime with implicit time-stepping schemes of arbitrary step size.

The computational cost or complexity of an AMG algorithm is typically measured in *work units* (WU), where one WU is the cost to perform one sparse matrix-vector multiplication with the initial matrix. Operator complexity (OC) gives the cost in WUs to perform one sparse matrix-vector multiplication on each level in an AMG hierarchy, and cycle complexity (CC) gives the cost in WUs to perform one AMG iteration, including pre- and postrelaxation, computing and restricting the residual, and coarse-grid correction. For convergence factor ρ, the *work-unit-per-digit-of-accuracy* (WPD), χ_{wpd}, is an objective measure of AMG performance, giving the total WUs necessary to achieve an order-of-magnitude reduction in the residual:[4]

$$\chi_{wpd} := -\frac{CC}{\log_{10}(\rho)} := -\frac{1}{\log_{10}(\rho_{eff})},$$

where $\rho_{eff} = \rho^{1/CC}$ is the *effective convergence factor*.

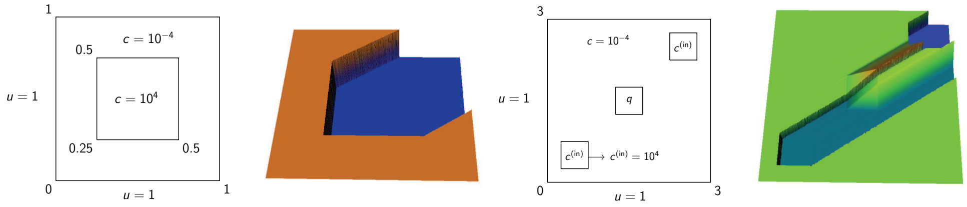
6.1. Test problems and discretizations.

The model problem used here is the steady state transport equation:

$$\begin{aligned} \mathbf{b}(x,y) \cdot \nabla u + c(x,y)u &= q(x,y), \quad && \mathcal{D}, \\ u &= g(x,y), && \Gamma_{\text{in}} \end{aligned} \tag{23}$$

[4] Although WPD is a good measure of serial performance, it does not reflect parallel efficiency of the algorithm.

for domain $\mathcal{D}$ and inflow boundary Γ_{in}. Multiple cases are studied that encompass spatially dependent source terms, $q(x,y)$, discontinuities in the material coefficient, $c(x,y)$, and constant and nonconstant flow direction, $\mathbf{b}(x,y)$, over structured and unstructured meshes. When $\mathbf{b}(x,y)$ is constant, we denote $\mathbf{b}(x,y) = \Omega(\theta) := (\cos(\theta), \sin(\theta))$, for some angle θ. Two model domains are considered the *inset* domain and *block-source* domain, shown with solutions in Figure 3. In each domain, inflow boundaries consist of the south and west boundaries with inflow $u = 1$, and the material coefficient $c(x,y)$ is piecewise constant in both cases, with changes of eight orders of magnitude, as shown in Figure 3. The *inset* domain has no source ($q = 0$), while the *block-source* domain has an interior source $q(x,y) = 1$ in the interior block. These terms ($c(x,y)$ and $q(x,y)$) are fixed for all experiments, except the nontriangular case (section 6.4). Multiple velocity fields $\mathbf{b}(x,y)$ are considered. Solutions with constant flow are shown in Figure 3, and several variations of the *inset* domain with nonconstant flow are shown in Figure 5 in section 6.2. All numerical experiments use $c(x,y)$ and $q(x,y)$ as specified in Figure 3, but multiple velocity fields $\mathbf{b}(x,y)$ are considered.



(a) *Inset* domain

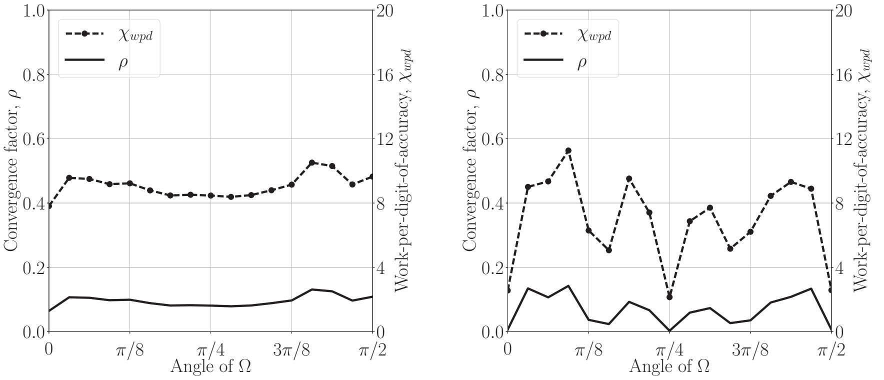
(b) *Block-source* domain

FIG. 3. *Two domains for the steady state transport equation, with a constant velocity field,* $\mathbf{b}(x,y) = (\cos(\theta), \sin(\theta))$, $\theta = 3\pi/16$, *source* $q = 1$, *and the respective solutions.*

To accompany the different domains considered, multiple upwind discretizations are implemented. A first-order lumped corner balance (LCB) finite element discretization [43, 44] is applied on structured and unstructured meshes. Standard fully upwinded DG discretizations [33, 52] are also tested, with finite element orders $1 - 6$. A comprehensive introduction can be found in [20]. Standard upwinded DG methods arise as special cases in [14] and for almost-scattering-free problems in [51]. The structured meshes used are triangular crossed-square meshes, conforming to the material discontinuities in $c(x,y)$, while random triangulations, again conforming to material discontinuities, are used as unstructured meshes. Additional discretizations based on highly elongated meshes with curvilinear elements, as well as continuous (linear) elements with artificial diffusion, are briefly explored in section 6.4.

As motivation for nAIR, we first highlight the difficulties that existing varieties of AMG face solving these discretizations. Tests were run using the PyAMG library [7], *hypre* [27], and ML [29]. Classical AMG methods are not well developed for the nonsymmetric setting; methods such as BoomerAMG in *hypre* [32] use a Galerkin, P^TAP coarse grid and are able to solve discontinuous transport discretizations based on linear finite elements, with convergence factors on the order of 0.8–0.9. However, convergence is not scalable and does not extend beyond linear elements. Aggregation- and energy-minimization-based AMG methods are better developed for nonsymmetric problems. The most successful existing solver appears to be the nonsymmetric smoothed aggregation (NSSA) solver in the ML library [29, 56, 61]. With GMRES acceleration, NSSA is able to converge on most problems tested here. In all cases, however, NSSA takes several times more iterations than nAIR and often requires significant relaxation, such as a $V(3,3)$-cycle, for good convergence. For difficult problems, nAIR offers a speedup of 5× or more over the current state of the art.

6.2. Angular variation, nonconstant flow, and three dimensions. Problems in higher dimensions, anisotropies on unstructured meshes, and non–grid-aligned anisotropies can prove difficult for AMG solvers [39, 57]. Here, we show nAIR to be robust in all of these cases. Figure 4 shows performance of nAIR for LCB discretizations of the *inset* problem on structured and unstructured meshes, with fixed angle $\Omega := \mathbf{b}(x,y) = (\cos(\theta), \sin(\theta))$ and angles $\theta \in [0, {}^{\pi}/_{2}]$. Because unstructured meshes are often used in practice and typically more difficult from a solver perspective, further results use an unstructured mesh and the angle is (arbitrarily) fixed to $\theta = {}^{3\pi}/_{16}$.[5]



(a) LCB unstructured mesh

(b) LCB structured mesh

FIG. 4. *Convergence factors and WPD for nAIR applied to LCB discretizations of the inset problem, with angles between* 0 *and* ${}^{\pi}/_{2}$, *on unstructured and structured meshes, and* $\approx 2.25M$ *DOFs.*

In addition to being robust with respect to angular variations, nAIR is insensitive to flow direction and problem dimensionality. Figure 5 shows the solution to three different nonconstant flows defined on the *inset* domain, and the corresponding performance of nAIR, along with results for a fixed direction on the inset and block-source domain.

Table 2 shows results of nAIR applied to steady state transport in three dimensions for different finite element orders. The three-dimensional domain is a unit cube, with $c(x,y) = 10^4$ inside of a centered interior cube of size $0.5 \times 0.5 \times 0.5$, and $c(x,y) = 10^{-4}$ outside of that subdomain, similar to the 2d *inset* domain (Figure 3). A random tetrahedral mesh is used, conforming to discontinuities in $c(x,y)$, and a constant velocity field, $\mathbf{b}(x,y) = (\sin(\theta_1)\cos(\theta_2), \sin(\theta_1)\sin(\theta_2), \cos(\theta_1)\cos(\theta_2))$. As in two dimensions, the choice of θ_1 and θ_2 do not affect results on an unstructured mesh (see Figure 4); here we use $\theta_1 = \theta_2 = 3\pi/16$. In all cases, nAIR is able to achieve fast convergence at a moderate cost. Due to the increased matrix connectivity in three dimensions, filtering is particularly useful here, reducing WPD by a factor of four or more in all cases, and total time-to-solution (not shown) by factors of 3–4.

[5] For some angles, nAIR converges faster on an unstructured mesh than a structured mesh. However, the wall-clock time of the setup and solve phase is at least 2× faster in *all cases* for a structured mesh over an unstructured mesh. It is possible that a structured mesh makes for a more structured matrix amenable to matrix-vector operations, but a detailed analysis is outside the scope of this work.

(a) $\mathbf{b}_1(x,y) = (\cos(\pi y)^2, \cos(\pi x)^2)$. (b) $\mathbf{b}_2(x,y) = (\sin(\pi y)^2, \sin(\pi x)^2)$. (c) $\mathbf{b}_3(x,y) = (y^4, \cos(^{\pi x}/_2)^2)$.

$\mathbf{b}(x,y)$	Ω_{inset}	$\mathbf{b}_1(x,y)$	$\mathbf{b}_2(x,y)$	$\mathbf{b}_3(x,y)$	$\Omega_{\text{block-source}}$
ρ	0.20	0.26	0.17	0.24	0.25
CC	6.83	6.77	6.80	6.81	7.53
χ_{WPD}	9.68	11.46	8.89	10.89	12.36

FIG. 5. *Convergence factor, CC, and WPD, for nAIR applied to variations in flow direction, $\mathbf{b}(x,y)$, on the inset domain, and constant flow direction $\Omega = (\cos(^{3\pi}/_{16}), \sin(^{3\pi}/_{16}))$ on both domains. All discretizations have $\approx$ nine million DOFs.*

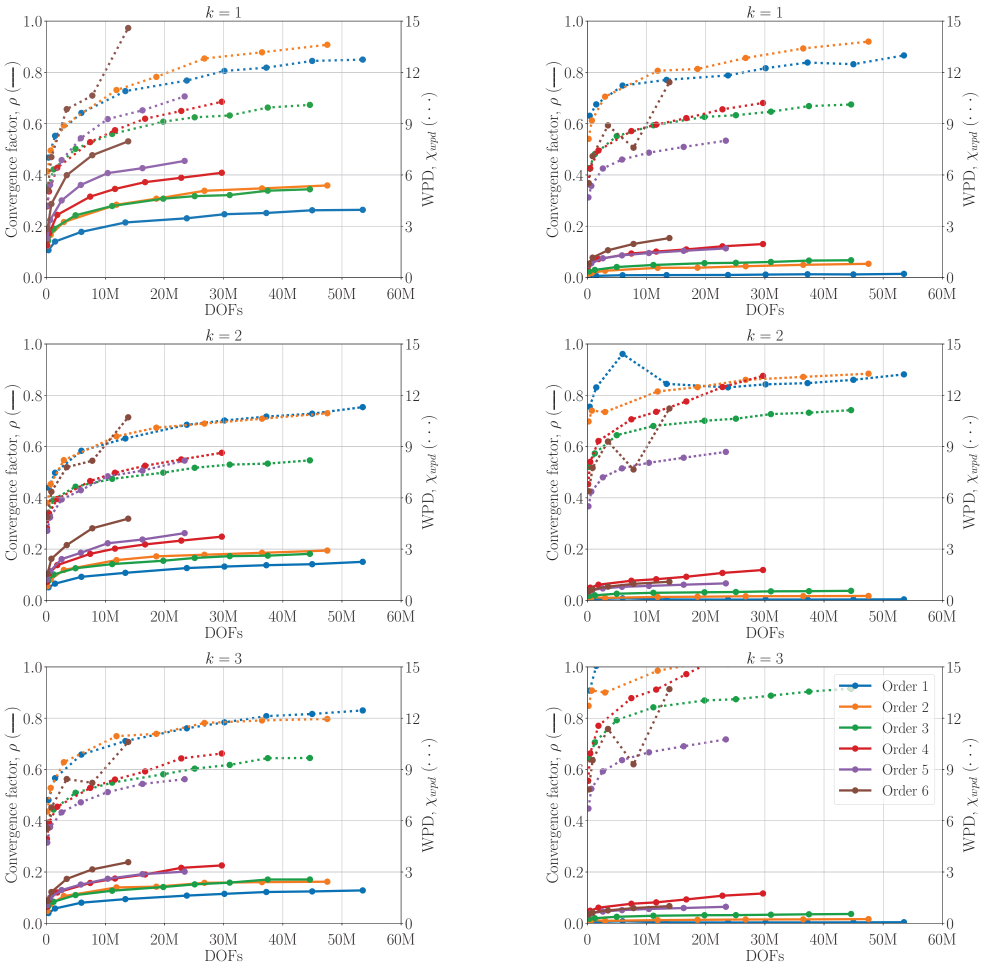
TABLE 2

(3D) nAIR applied to first-, second-, and third-order discretizations of steady state transport in three dimensions. The final column shows the speedup due to filtering in terms of WPD. The rows in each block differ in drop tolerance, φ.

Degree FEM	n	nnz	ρ	CC	χ_{WPD}	φ	Speedup
1	2.5M	24M	0.09	42.4	41.7	0	–
1	2.5M	24M	0.10	10.9	11.0	1e-3	3.9
2	1.9M	41M	0.12	38.4	42.8	0	–
2	1.9M	41M	0.13	8.5	9.9	1e-3	4.3
3	2.2M	85M	0.14	32.6	38.2	0	–
3	2.2M	85M	0.17	6.8	8.9	1e-3	4.3

This is also a good example that demonstrates the speedup that nAIR can provide over ℓAIR in setup time. For third-order elements in three dimensions, with about two million DOFs, distance-1 ℓAIR takes 2995 seconds to build the solver and 43 seconds to solve to 10^{-12} residual tolerance. Distance-one nAIR (approximately corresponding to distance-two ℓAIR) takes only 38 seconds to setup and 52 seconds to solve. Despite convergence factors that are slightly larger with nAIR compared to ℓAIR, the overall time to solution is significantly smaller. Furthermore, it is simple and moderately cheap to go from distance-one to distance-two nAIR; for example, the setup time here increases modestly to about 60 seconds with distance-two nAIR, while distance-three ℓAIR (approximately corresponding to distance-two nAIR) is completely intractable.

6.3. Scaling in h and element order. Next, we study the scaling of nAIR with respect to DOFs. One exciting feature of nAIR is its ability to solve high-order finite-element discretizations, something that AMG methods often struggle with. Here, V-cycles and F-cycles are considered, with degree of nAIR $k = 1, 2$, and 3. Although F-cycles originate in full multigrid, which focuses on achieving discretization-level accuracy in a single F-cycle [15], accuracy with respect to discretization is not considered in this work. Instead, the F-cycle is used as it can provide more robust convergence and scaling than a V-cycle, at a much lower cost than alternatives such as W-cycles and K-cycles. Figure 6 shows scaling of WPD and convergence factor of nAIR applied to upwind DG discretizations of the inset problem as a function of the number of DOFs. Although there remains a slow growth in WPD, likely due to an increase in iterations before asymptotic convergence rates are achieved, convergence factors have effectively asymptoted for lower-order finite elements, and are leveling off even for sixth-order finite elements.



(a) V-cycle

(b) F-cycle

FIG. 6. *Scaling of convergence factor (solid lines) and WPD (dotted lines) as a function of DOFs, for nAIR applied to upwind DG discretizations of the inset problem, with nAIR degrees* $k = 1, 2,$ *and* 3*, V-cycles and F-cycles, and finite-element degrees* 1*–*6*.*

One benefit of the reduction approach is that convergence of nAIR can be improved by increasing the accuracy of the Neumann approximation. Figure 6 demonstrates that convergence is improved a notable amount by increasing $k = 1$ to $k = 2$, and again for $k = 2$ to $k = 3$. This allows us to attain V-cycle convergence factors on the order of $\rho \approx 0.2$ for sixth-order finite elements. Similar improvements in convergence can be obtained by decreasing the strength tolerance, ϕ, leading to a more accurate approximation of A_{ff}^{-1}. Because increasing k also increases the density of coarse-grid operators, in practice, one must find the balance between complexity and convergence factors. Here, we see that the V-cycle with $k = 2$ appears to be the most effective choice, because the WPD is less than that of $k = 1, 3$, or F-cycles for all finite element orders. The best choice in terms of total time (including setup) likely depends on how many linear systems are being solved, and it is possible $k = 1$ is faster for a single or small number of systems. Note that having the option to make this choice is a benefit of AIR, because, in general, classical AMG methods do not have a natural and robust way to improve convergence the way that increasing k can.

So far all results have employed scaling the matrix by the block-diagonal inverse. However, nAIR is also amenable to a block implementation, where coarsening, restriction, and interpolation are all done by block. This is particularly relevant for systems of PDEs with coupled variables, where scaling out the block diagonal is unlikely to capture the necessary couplings. Figure 7 demonstrates block nAIR applied to the same problem as in Figure 6, this time directly using the DG block structure, for orders 1, 2, and 3 finite elements. Convergence factors are similar to those achieved by scaling by the block diagonal inverse. The WPD is higher because we do not filter in the block setting, and because, when using a block sparse matrix, even zero entries in otherwise nonzero blocks must be stored. This results in a larger OC and, thus, larger WPD.

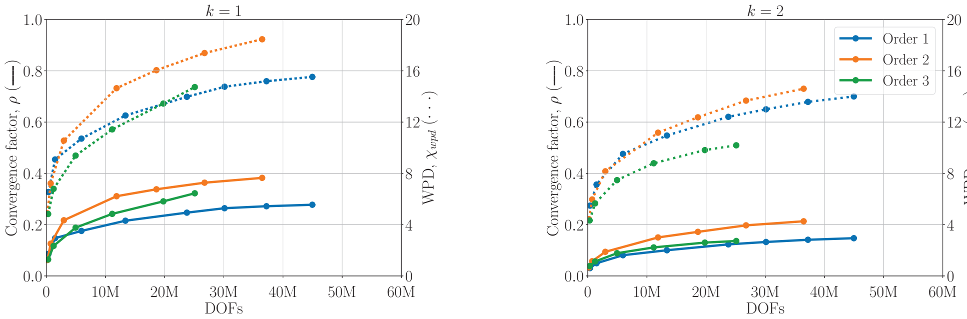


FIG. 7. *Scaling of convergence factor (solid lines) and WPD (dotted lines) as a function of DOFs, for block-nAIR applied to upwind DG discretizations of the inset problem, with nAIR degrees $k = 2$ and 3, V-cycles, finite element degrees 1–3, and corresponding block sizes 3, 6, and 10.*

6.4. Nontriangular matrices. Finally, it is well known that a Neumann approximation for ideal operators is, in general, not effective for nontriangular matrices, such as a symmetric discretization of diffusion. However, for "nearly triangular" matrices, nAIR remains an effective solver. Here, we demonstrate the performance of nAIR on three such problems in Table 3.

P1 corresponds to a streamline upwind Petrov–Galerkin (SUPG) discretization of (23) on the *inset* problem, using linear finite elements. SUPG discretizations use an upwinded scheme for advection, with a small diffusive component added for stability [16]. This results in a small, global, symmetric component added to a triangular matrix. P2 and P3 correspond to fourth-order, upwind DG discretizations of (23) in two dimensions on high-order curvilinear finite elements [62]. Here, $c(x, y) = 2xy + 2x^2 + 1.2$, $\mathbf{b}(x, y) = (1/\sqrt{3}, 1/\sqrt{3})$, and $q(x, y)$ is the right-hand side corresponding to the exact solution $u = (x^2 + y^2 + 1)/2 + \cos(2.5(x + y))(b_0^2 + b_1)$. Curvilinear finite elements can be nonconvex and produce cycles in the mesh, wherein the resulting matrix for a fixed direction of flow is mostly block triangular, with some number of strongly connected components (SCCs) that are nontriangular and large in magnitude [30]. P2 has 97 SCCs of size two and 15 SCCs of 3–6 DOFs. P3 has 40 SCCs of size two, 11 small SCCs with 3–6 DOFs, and one large SCC consisting of 1951 DOFs, implying there is a substantial nontriangular block in the matrix.

Table 3

Results of nAIR-preconditioned GMRES applied to the nearly triangular discretizations P1, P2, and P3, corresponding to an SUPG discretization of the inset problem, and 4th-order DG discretizations of a variation in (23) on two meshes with curvilinear elements, respectively (see text for details).

Problem	n	k	ρ	CC	χ_{WPD}
P1	1553001	1	0.57	8.56	35.74
P2	88800	1	0.22	4.52	6.85
P2	88800	2	0.14	5.63	6.71
P3	88800	1	0.46	4.54	13.59
P3	88800	2	0.35	5.72	12.62

It should be noted that even in serial, solving such a problem algebraically is nontrivial, particularly when there is a global symmetric component. One approach is to use a lower-triangular preconditioner that inverts the advective components exactly, equivalent to an ordered Gauss–Seidel iteration. This would likely converge well; however, without geometric knowledge of the velocity field and corresponding ordering of DOFs in the matrix, the right relaxation ordering cannot be easily determined. In [30], a cycle-breaking strategy was proposed to approximate an optimal relaxation ordering in the context of larger transport simulations on meshes with curvilinear elements. This proved successful on linear systems with SCCs, but convergence of the larger transport "source iteration" was $2-3\times$ slower than using an exact solve, and this approach does not overcome the inherent limitation of Gauss–Seidel in parallel.

Each of these problems is nearly triangular in a different regard. Here, we apply one nAIR V-cycle as a preconditioner for GMRES, and results in Table 3 show nAIR to be an effective preconditioner in all cases.

Classical AMG with GMRES acceleration does not converge on either problem. NSSA using GMRES acceleration and Jacobi relaxation converges for both problems. For the harder problem, P_3, NSSA converges with factors on the order of $\rho = 0.78$ for a $V(2,2)$-cycle and $\rho = 0.68$ for a $V(3,3)$-cycle, with cycle complexities likely on the order of 8, 9, and 11, respectively, and corresponding WPDs of 66 and 79.[6] However, this convergence is sensitive to parameters. For example, reasonable convergence is attained for an aggregation threshold of $\phi = 0.2$, but NSSA does not converge for $\phi = 0.3$, and iterations are halted due to a singular GMRES Hessenberg matrix for $\phi = 0.1$.

7. Conclusions.

This work studies reduction-based AMG for highly nonsymmetric matrices. Theory is developed indicating that, along with accurate approximations of the ideal operators, a scalable method also requires that interpolation or restriction satisfy a classical multigrid approximation property. A reduction-based AMG method is then developed, denoted nAIR, which is shown to be an effective solver for upwind discretizations. Strong convergence factors are shown when nAIR

[6] ML does not detail complexity in the same manner as PyAMG, so these estimates are for comparison purposes between NSSA and nAIR.

is applied to the steady state transport equation, on multiple domains, with high-order upwind discretizations, and unstructured meshes or meshes with curvilinear elements. Although nAIR as presented here proves robust for several "nearly triangular" problems, when significant nontriangular components are introduced, the performance of nAIR will quickly degrade, and the more general variation, ℓAIR [58], is more appropriate.

A serial implementation of nAIR is available in the PyAMG library [7], available at https://github.com/ben-s-southworth/pyamg/, and a parallel implementation is now available in the *hypre* library [27] at https://github.com/hypre-space/hypre.

Acknowledgment. The authors thank Steve McCormick for helpful discussions in the development of nAIR.

REFERENCES

[1] M. P. Adams, M. L. Adams, and W. D. Hawkins, *Provably optimal parallel transport sweeps on regular grids*, in International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering, Sun Valley, Idaho, 2013, pp. 2535–2553.

[2] F. L. Alvarado and R. Schreiber, *Optimal parallel solution of sparse triangular systems*, SIAM J. Sci. Comput., 14 (1993), pp. 446–460, https://doi.org/10.1137/0914027.

[3] S. Amarala and J. W. L. Wan, *Multigrid methods for systems of hyperbolic conservation laws*, Multiscale Model. Simul., 11 (2013), pp. 586–614, https://doi.org/10.1137/110851316.

[4] E. Anderson and Y. Saad, *Solving sparse triangular linear systems on parallel computers*, International Journal of High Speed Computing, 1 (1989), pp. 73–95.

[5] T. S. Bailey and R. D. Falgout, *Analysis of massively parallel discrete-ordinates transport sweep algorithms with collisions*, in International Conference on Mathematics, Computational Methods, and Reactor Physics, Saratoga Springs, NY, 2009, Lawrence Livermore National Laboratory, Livermore, pp. 1751–1765.

[6] A. H. Baker, R. D. Falgout, T. V. Kolev, and U. M. Yang, *Scaling Hypre's Multigrid Solvers to* 100,000 *Cores*, in High-Performance Scientific Computing, Springer, London, 2012, pp. 261–279.

[7] W. N. Bell, L. N. Olson, and J. B. Schroder, *PyAMG: Algebraic Multigrid Solvers in Python v*3.0, release 3.0, https://github.com/pyamg/, 2015.

[8] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[9] A. Bienz, R. D. Falgout, W. Gropp, L. N. Olson, and J. B. Schroder, *Reducing parallel communication in algebraic multigrid through sparsification*, SIAM J. Sci. Comput., 38 (2016), pp. S332–S357, https://doi.org/10.1137/15M1026341.

[10] A. Brandt, S. F. McCormick, and J. Ruge, *Algebraic multigrid (AMG) for sparse matrix equations*, Sparsity and its Applications, Cambridge University Press, Cambridge, 1985, pp. 257–284.

[11] J. Brannick, F. Cao, K. Kahl, R. D. Falgout, and X. Hu, *Optimal interpolation and compatible relaxation in classical algebraic multigrid*, SIAM J. Sci. Comput., 40 (2018), pp. A1473–A1493, https://doi.org/10.1137/17M1123456.

[12] J. J. Brannick, A. Frommer, K. Kahl, S. P. MacLachlan, and L. T. Zikatanov, *Adaptive reduction-based multigrid for nearly singular and highly disordered physical systems*, Electron. Trans. Numer. Anal., 37 (2010), pp. 276–295.

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

[14] F. Brezzi, L. D. Marini, and E. Süli, *Discontinuous Galerkin methods for first-order hyperbolic problems*, Math. Models Methods Appl. Sci., 14 (2004), pp. 1893–1903.

[15] W. L. Briggs, V. E. Henson, and S. F. McCormick, 2nd ed., *A Multigrid Tutorial*, SIAM, Philadelphia, 2000, https://doi.org/10.1137/1.9780898719505.

[16] A. N. Brooks and T. J. Hughes, *Streamline upwind Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg., 32 (1982), pp. 199–259.

[17] L. M. Carvalho, L. Giraud, and P. Le Tallec, *Algebraic two-level preconditioners for the Schur complement method*, SIAM J. Sci. Comput., 22 (2001), pp. 1987–2005, https://doi.org/10.1137/S1064827598340809.

[18] G. Colomer, R. Borrell, F. X. Trias, and I. Rodríguez, *Parallel algorithms for SN transport sweeps on unstructured meshes*, J. Comput. Phys., 232 (2013), pp. 118–135.

[19] H. De Sterck, R. D. Falgout, J. W. Nolting, and U. M. Yang, *Distance-two interpolation for parallel algebraic multigrid*, Numer. Linear Algebra Appl., 15 (2008), pp. 115–139.

[20] D. Di Pietro and A. Ern, *Mathematical Aspects of Discontinuous Galerkin Methods*, Mathématiques et Applications, Springer, Heidelberg, 2012.

[21] V. Dobrev, T. Kolev, N. A. Petersson, and J. B. Schroder, *Two-level convergence theory for multigrid reduction in time (MGRIT)*, SIAM J. Sci. Comput., 39 (2017), pp. S501–S527, https://doi.org/10.1137/16M1074096.

[22] R. D. Falgout, T. A. Manteuffel, B. O'Neill, and J. B. Schroder, *Multigrid reduction in time for nonlinear parabolic problems: A case study*, SIAM J. Sci. Comput., 39 (2017), pp. S298–S322, https://doi.org/10.1137/16M1082330.

[23] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., 36 (2014), pp. C635–C661, https://doi.org/10.1137/130944230.

[24] R. D. Falgout and J. E. Jones, *Multigrid on massively parallel architectures*, in Multigrid Methods, VI (Gent, 1999), Lect. Notes Comput. Sci. Eng. 14, Springer, Berlin, 2000, pp. 101–107.

[25] R. D. Falgout and J. B. Schroder, *Non-Galerkin coarse grids for algebraic multigrid*, SIAM J. Sci. Comput., 36 (2014), pp. C309–C334, https://doi.org/10.1137/130931539.

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

[27] R. D. Falgout and U. M. Yang, *HYPRE: A library of high performance preconditioners*, European Conference on Parallel Processing, Lecture Notes in Comput. Sci. 2331, Springer, Berlin, Heidelberg, 2002, pp. 632–641.

[28] H. Foerster, K. Stüben, and U. Trottenberg, *Non-standard multigrid techniques using checkered relaxation and intermediate grids*, in Elliptic Solvers, Academic Press, New York, 1981, pp. 285–300.

[29] M. Gee, C. Siefert, J. Hu, R. Tuminaro, and M. Sala, *ML* 5.0 *Smoothed Aggregation User's Guide*, Tech. Report SAND2006-2649, Sandia National Laboratories, 2006.

[30] T. S. Haut, P. G. Maginot, V. Z. Tomov, B. S. Southworth, T. A. Brunner, and T. S. Bailey, *An efficient sweep-based solver for the S_N equations on high-order meshes*, Nucl. Sci. Eng., 193 (2019), pp. 746–759.

[31] W. Hawkins, *Efficient massively parallel transport sweeps*, in Transactions of the American Nuclear Society 107, Texas A & M University, College Station, TX, 2012, pp. 477–481.

[32] V. E. Henson and U. M. Yang, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177.

[33] P. Lesaint and P. Raviart, *On a finite element method for solving the neutron transport equation*, Mathematical Aspects of Finite Elements in Partial Differential Equations, C. de Boor, ed., Academic Press, New York, 1974, pp. 89–123.

[34] R. Li and Y. Saad, *GPU-accelerated preconditioned iterative linear solvers*, J. Supercomput., 63 (2013), pp. 443–466.

[35] W. Liu, A. Li, J. Hogg, I. S. Duff, and B. Vinter, *A synchronization-free algorithm for parallel sparse triangular solves*, in European Conference on Parallel Processing, Lecture Notes in Comput. Sci. 9833, Springer, Cham, 2016, pp. 617–630.

[36] J. Lottes, *Towards Robust Algebraic Multigrid Methods for Nonsymmetric Problems*, Springer, Cham, 2017.

[37] S. P. MacLachlan, T. A. Manteuffel, and S. F. McCormick, *Adaptive reduction-based AMG*, Numer. Linear Algebra Appl., 13 (2006), pp. 599–620.

[38] J. Mandel, *On block diagonal and Schur complement preconditioning*, Numer. Math., 58 (1990), pp. 79–93.

[39] T. A. Manteuffel, L. N. Olson, J. B. Schroder, and B. S. Southworth, *A root-node based algebraic multigrid method*, SIAM J. Sci. Comput., 39 (2017), pp. S723–S756, https://doi.org/10.1137/16M1082706.

[40] T. Manteuffel and B. S. Southworth, *Convergence in norm of nonsymmetric algebraic multigrid*, SIAM J. Sci. Comput., 41 (2019), pp. S269–S296, https://doi.org/10.1137/18M1193773.

[41] C. Mense and R. Nabben, *On algebraic multi-level methods for non-symmetric systems–*

Comparison results, Linear Algebra Appl., 429 (2008), pp. 2567–2588.

[42] C. Mense and R. Nabben, *On algebraic multilevel methods for non-symmetric systems—convergence results*, Electron. Trans. Numer. Anal., 30 (2008), pp. 323–345.

[43] J. E. Morel and J. S. Warsa, *An S_n spatial discretization scheme for tetrahedral meshes*, Nucl. Sci. Eng., 151 (2005), pp. 157–166.

[44] J. E. Morel and J. S. Warsa, *Spatial finite-element lumping techniques for the quadrilateral mesh S_n equations in X-Y geometry*, Nucl. Sci. Eng., 156 (2007), pp. 325–342.

[45] Y. Notay, *A robust algebraic multilevel preconditioner for non-symmetric M-matrices*, Numer. Linear Algebra Appl., 7 (2000), pp. 243–267.

[46] Y. Notay, *Algebraic analysis of two-grid methods: The nonsymmetric case*, Numer. Linear Algebra Appl., 17 (2010), pp. 73–96.

[47] Y. Notay, *Analysis of Two-Grid Methods: The Nonnormal Case*, Tech. Report GANMN 18-01, 2018.

[48] L. N. Olson, J. B. Schroder, and R. S. Tuminaro, *A general interpolation strategy for algebraic multigrid using energy minimization*, SIAM J. Sci. Comput., 33 (2011), pp. 966–991, https://doi.org/10.1137/100803031.

[49] C. W. Oosterlee, F. J. Gaspar, T. Washio, and R. Wienands, *Multigrid line smoothers for higher order upwind discretizations of convection-dominated problems*, J. Comput. Phys., 139 (1998), pp. 274–307.

[50] H. Park, D. A. Knoll, R. M. Rauenzahn, C. K. Newman, J. D. Densmore, and A. B. Wollaber, *An efficient and time accurate, moment-based scale-bridging algorithm for thermal radiative transfer problems*, SIAM J. Sci. Comput., 35 (2013), pp. S18–S41, https://doi.org/10.1137/120881075.

[51] J. C. Ragusa, J. L. Guermond, and G. Kanschat, *A robust SN-DG-approximation for radiation transport in optically thick and diffusive regimes*, J. Comput. Phys., 231 (2012), pp. 1947–1962.

[52] W. Reed and T. Hill, *Triangular Mesh Methods for the Neutron Transport Equation*, Tech. Rep. LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, NM, 1973.

[53] M. Ries, U. Trottenberg, and G. Winter, *A note on MGR methods*, Linear Algebra Appl., 49 (1983), pp. 1–26.

[54] J. W. W. Ruge and K. Stüben, *Algebraic multigrid*, Multigrid Methods, Frontiers Appl. Math. 3, SIAM, Philadelphia, 1987, pp. 73–130.

[55] Y. Saad and M. Sosonkina, *Distributed Schur complement techniques for general sparse linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 1337–1356, https://doi.org/10.1137/S1064827597328996.

[56] M. Sala and R. S. Tuminaro, *A new Petrov–Galerkin smoothed aggregation preconditioner for nonsymmetric linear systems*, SIAM J. Sci. Comput., 31 (2008), pp. 143–166, https://doi.org/10.1137/060659545.

[57] J. B. Schroder, *Smoothed aggregation solvers for anisotropic diffusion*, Numer. Linear Algebra Appl., 19 (2012), pp. 296–312.

[58] T. A. Manteuffel, J. W. Ruge and B. S. Southworth, *Nonsymmetric algebraic multigrid based on local approximate ideal restriction (ℓAIR)*, SIAM J. Sci. Comput., 40 (2018), pp. A4105–A4130, https://doi.org/10.1137/17M1144350.

[59] E. Treister and I. Yavneh, *Non-Galerkin multigrid based on sparsified smoothed aggregation*, SIAM J. Sci. Comput., 37 (2015), pp. A30–A54, https://doi.org/10.1137/140952570.

[60] P. S. Vassilevski, *Multilevel Block Factorization Preconditioners*, in Matrix-based Analysis and Algorithms for Solving Finite Element Equations, Springer, New York, 2008.

[61] T. A. Wiesner, R. S. Tuminaro, W. A. Wall, and M. W. Gee, *Multigrid transfers for nonsymmetric systems based on Schur complements and Galerkin projections*, Numer. Linear Algebra Appl., 21 (2014), pp. 415–438.

[62] D. N. Woods, T. A. Brunner, and T. S. Palmer, *High order finite element S_n transport in $x-y$ geometry on meshes with curved surfaces*, Trans. Am. Nucl. Soc., 114 (2016), pp. 377–380.

[63] X. Xu and C.-S. Zhang, *On the ideal interpolation operator in algebraic multigrid methods*, SIAM J. Numer. Anal., 56 (2018), pp. 1693–1710, https://doi.org/10.1137/17M1162779.

[64] I. Yavneh, C. H. Venner, and A. Brandt, *Fast multigrid solution of the advection problem with closed characteristics*, SIAM J. Sci. Comput., 19 (1998), pp. 111–125, https://doi.org/10.1137/S1064827596302989.