TOWARDS ADAPTIVE SMOOTHED AGGREGATION (αSA) FOR NONSYMMETRIC PROBLEMS *

M. BREZINA , T. MANTEUFFEL , S. MCCORMICK , J. RUGE , AND G. SANDERS†

Abstract. Applying smoothed aggregation multigrid (SA) to solve a nonsymmetric linear system, $A\mathbf{x} = \mathbf{b}$, is often impeded by the lack of a minimization principle that can be used as a basis for the coarse-grid correction process. This paper proposes a Petrov-Galerkin (PG) approach based on applying SA to either of two symmetric positive definite (SPD) matrices, $\sqrt{A^tA}$ or $\sqrt{AA^t}$. These matrices, however, are typically full and difficult to compute, so it is not computationally efficient to use them directly to form a coarse-grid correction. The proposed approach approximates these coarse-grid corrections by using SA to accurately approximate the right and left singular vectors of A that correspond to the lowest singular value. These left and right singular vectors are used to construct the restriction and interpolation operators, respectively. Some preliminary two-level convergence theory is presented, suggesting more relaxation should be applied than for a SPD problem. Additionally, a nonsymmetric version of adaptive SA (αSA) is given that automatically constructs SA multigrid hierarchies using a stationary relaxation process on all levels. Numerical results are reported for convection-diffusion problems in two-dimensions with varying amounts of convection for constant, variable, and recirculating convection fields. The results suggest that the proposed approach is algorithmically scalable for problems coming from these nonsymmetric scalar PDEs (with the exception of recirculating flow). This paper serves as a first step for nonsymmetric αSA. The long-term goal of this effort is to develop nonsymmetric αSA for systems of PDEs, where the SA framework has proven well-suited for adaptivity in SPD problems.

Key words. smoothed aggregation, algebraic multigrid, nonsymmetric, adaptive, USYMQR, Petrov-Galerkin

AMS subject classifications.

1. Introduction.

Consider solving a system of linear equations,

$$A\mathbf{x} = \mathbf{b}, \tag{1.1}$$

where A is a large, real, sparse, nonsingular, and possibly *nonsymmetric* ($A \neq A^t$) matrix of size $n \times n$, $\mathbf{x}$ is an unknown vector, and $\mathbf{b}$ is a known vector. Additionally, assume $\mathrm{Re}\,\langle A\mathbf{x}, \mathbf{x}\rangle > 0$. Problems such as this are commonly utilized to obtain numerical solutions to steady-state, time-dependent, or nonlinear PDEs. For many applications, the computational time spent on linear problems dominates the total simulation time. The goal of the methods presented in this paper is to automatically form an iterative method that solves (1.1) in a computationally efficient way, without requiring information regarding the origin of the problem.

For a large class of these linear systems, multigrid methods provide optimal solvers ([4], see [11, 24] for introduction). The favorable convergence properties of these methods stem from combining two complementary error-reduction processes: a local *relaxation* process, and a *coarse-grid correction*, given in stationary two-grid form as

$$\mathbf{x} \leftarrow \mathbf{x} + P(R^tAP)^{-1}R^t\mathbf{r}, \qquad \mathbf{r} := \mathbf{b} - A\mathbf{x}, \tag{1.2}$$

with intergrid transfer operators R^t (restriction) and P (interpolation). (Throughout this paper, we depart from the usual notation by using the transpose, R^t, to denote restriction.) Ideally, the relaxation process efficiently attenuates much of the error, with

*Submitted to the SIAM Journal on Scientific Computing, June 15, 2008

†Department of Applied Mathematics, Campus Box 526, University of Colorado at Boulder, Boulder, CO, 80309-0526. *email:* mbrezina@math.cudenver.edu, tmanteuf@colorado.edu, stevem@colorado.edu, jruge@colorado.edu, and sandersg@colorado.edu.

the remaining error referred to as *algebraically smooth error*. If coarse spaces with much smaller dimension and sparse bases can be constructed to accurately represent the algebraically smooth error, then an adequate two-grid method results. Approximation of $(R^tAP)^{-1}$ by recursive application of relaxation and coarse-grid correction results in an efficient multigrid method, provided the sparsity of coarse problem matrices is controlled. Choosing a relaxation with good smoothing properties and constructing coarse subspaces with adequate approximation properties (and the associated multigrid intergrid transfer operators) are the general goals when designing a multigrid method.

Geometric multigrid methods assume that algebraically smooth error is also geometrically smooth. For certain nonsymmetric problems, choosing a specialized relaxation can meet this assumption. In [31, 30], it is shown that using a sequential relaxation method with a down-wind ordering and geometric coarsening yields an efficient method for two-dimensional convection diffusion problems, including recirculating flows. One caveat here is that the relaxation method used in this approach is sequential, and the orderings are reported to be integral in the success of these methods. Such orderings may be complicated to automatically calculate for a general problem and there is potential difficulty to successfully parallelize sequential relaxation methods. For these reasons, this work seeks to form an optimal multigrid method that does not rely on an order-dependent, sequential relaxation. Instead, we use nonstationary relaxation (USYMQR [22]) that only involves matrix-vector multiplies with A and A^t. The error that is algebraically smooth with respect to this relaxation is not geometrically smooth for the general problem and, therefore, we require a more general approach to coarse grid construction.

Classical algebraic multigrid (AMG [5, 6, 19, 24]) assumes that algebraically smooth error is geometrically smooth in directions of strong coupling, determined by the graph of the problem matrix. However, such assumptions do not hold for many problems of interest, limiting the usefulness of these approaches. Classical AMG has been successfully applied to two-dimensional convection diffusion problems with many types of flow [19], where these assumptions hold. The AMG coarse-grid selection successfully semi-coarsens in portions of the domain where algebraically smooth error tends to be smooth only in certain directions, and coarsens normally in other locations.

The coarsening approach proposed in this paper makes no assumption regarding the geometric smoothness of algebraically smooth error. Instead, this approach focuses on constructing coarse spaces within the smoothed aggregation (SA [25, 27, 28]) framework. Our SA approach assumes a certain type of relaxation process, based on residual correction of the form

$$\mathbf{x} \leftarrow \mathbf{x} + M^{-1}\mathbf{r}, \tag{1.3}$$

where M^{-1} is an inexpensive local approximation to A^{-1}. For such relaxation processes, error vectors that are *near-kernel components* (NK) in the sense that

$$\mathbf{e} \neq \mathbf{0} \qquad \text{such that} \qquad \frac{\|A\mathbf{e}\|}{\|\mathbf{e}\|} \approx \min_{\mathbf{v}\neq\mathbf{0}} \frac{\|A\mathbf{v}\|}{\|\mathbf{v}\|}, \tag{1.4}$$

are also algebraically smooth and the terms are commonly used interchangeably. (Throughout this paper, the ℓ_2 norm is written $\|\mathbf{z}\| = \sqrt{<\mathbf{z},\,\mathbf{z}>}$ and matrix-induced vector norms are written $\|\mathbf{z}\|_B = \sqrt{<B\mathbf{z},\mathbf{z}>}$.) Interpolation operators are designed

to have good approximation for NK vectors. SA is particularly attractive for symmetric problems obtained by discretizing systems of PDEs, which tend to have richer sets of NK components and, thus, require interpolation operators that accommodate such error. The SA framework does so in a more systematic way than classical AMG. Given a set of algebraically smooth error prototypes, SA forms a multigrid hierarchy with adequate approximation to the entire set. For this reason, we wish to extend the SA framework to nonsymmetric problems.

Much of the design of multigrid methods within the literature relies on the problem matrix, A, being symmetric positive-definite (SPD). For such problems, a method using Galerkin or variational coarse-grid correction (where $R = P$) has a two-grid error propagation operator that is an energy-orthogonal projection. Thus, coupled with a relaxation method that is convergent in the energy norm, the two-grid method is guaranteed to be convergent. However, Galerkin coarsening does not guarantee convergence for nonsymmetric problems. The error propagation operator of Galerkin two-grid correction with nonsymmetric A is an *oblique projection* in the sense that the spaces involved are not orthogonal with respect to any known, practical inner product. Finding an inner product in which these spaces are orthogonal may require more computation than the solution to the original problem (1.1), or may not be possible. Furthermore, designing a relaxation that is convergent in this unknown inner product seems an equally daunting task. For this reason, we lift the Galerkin coarsening restriction and allow Petrov-Galerkin (PG) coarsening, formed with restriction and interpolation operators that are not equal.

Versions of SA with PG coarse grids have already been developed for nonsymmetric problems. Instead of merely representing NK on coarse-grids, these PG approaches represent both *left and right near-kernel components* (LRNK). Restriction is designed to accurately represent *left near-kernel components* (LNK), taken here to mean

$$\mathbf{e} \neq \mathbf{0} \qquad \text{such that} \qquad \frac{\|A^t\mathbf{e}\|}{\|\mathbf{e}\|} \approx \min_{\mathbf{u}\neq\mathbf{0}} \frac{\|A^t\mathbf{u}\|}{\|\mathbf{u}\|}, \tag{1.5}$$

and interpolation is designed to accurately represent *right near-kernel components* (RNK), which we take to be the same as our definition of near-kernel in (1.4). In [13], convective parts and diffusive parts of the problem matrix are coarsened separately, an approach that is not obviously applicable to a general problem. In [20], nonsymmetric linear systems are solved by using a PG solver hierarchy as a preconditioner for implicitly restarted GMRES. This method assumes the constant vector to be an adequate representation of for both LNK and RNK and uses it to build a tentative interpolation operator. The main feature of [20] is that it employs a different intergrid transfer operator smoothing for each column of interpolation, and a different smoothing for each row of restriction. On a vector-by-vector basis, columns of interpolation are individually smoothed to better approximate RNK on coarse grids, while rows of restriction are smoothed individually to better approximate LNK.

To provide motivation for our approach, we first consider existing convergence theory for multigrid [3, 16] and SA [26] for SPD problems with variational coarse grids based on either of the following approximation properties.

ASSUMPTION 1.1. (SYMMETRIC WEAK APPROXIMATION PROPERTY): An interpolation operator, P, satisfies the *weak approximation property* with constant K_w if, for any $\mathbf{e}$ on the fine grid, there exists an $\mathbf{e}_c$ on the coarse grid such that

$$\|\mathbf{e} - P\mathbf{e}_c\|^2 \leq \frac{K_w}{\|A\|} \langle A\mathbf{e}, \mathbf{e}\rangle . \tag{1.6}$$

ASSUMPTION 1.2. (SYMMETRIC STRONG APPROXIMATION PROPERTY): An interpolation operator, P, satisfies the *strong approximation property* with constant K_s if, for any $\mathbf{e}$ on the fine grid, there exists an $\mathbf{e}_c$ on the coarse grid such that

$$\|\mathbf{e} - P\mathbf{e}_c\|_A^2 \leq \frac{K_s}{\|A\|} \langle A\mathbf{e},\, A\mathbf{e}\rangle\,, \tag{1.7}$$

where $\|\,.\,\|_A$ is the well-known energy norm. Generally, if either one of these approximation properties holds on all levels of the multigrid hierarchy with constants that are bounded by K (with some additional assumptions), then convergence of the multigrid hierarchy is bounded by $1-\mathcal{O}(K^{-1})$. The weak approximation property is easier to enforce locally here, so theory that assumes the weak approximation property is preferred to theory that assumes the strong approximation property. However, additional assumptions may be required for convergence when the weak approximation property is utilized. Some theory has been developed for nonsymmetric problems for multigrid methods and variational coarse grids in [1, 2, 15, 29]. Instead, we develop a nonsymmetric generalization to the strong approximation property for PG coarse grids that, with an additional assumption, guarantees two-grid convergence with a sufficient amount of relaxation.

Previously, adaptive smoothed aggregation (αSA [9]) was applied to symmetric applications where a representative set of NK vectors is neither obvious nor supplied. First, a *primary near-kernel* vector is developed by applying a multilevel version of relaxation to the homogeneous problem, $A\mathbf{x} = \mathbf{0}$. The resulting initial vector is used to create a SA multigrid solver hierarchy. The solver hierarchy is used in place of relaxation to test the current method on the homogeneous problem. If convergence is inadequate, then the remaining error is used as a *secondary near-kernel* vectors. The two near-kernel vectors are used to create a new SA multigrid solver hierarchy that attempts to satisfy the weak approximation property locally with a smaller approximation constant. The process is repeated: test the current solver and develop a better NK representation if the solver is not yet adequate. αSA has been employed to develop optimal solvers for symmetric (or Hermitian) problems with applications to quantum dynamics [7], linear elasticity [28], and other applications involving systems PDEs. Although no provisions are made here for systems of PDEs, the accommodation of such problems is the long term goal of the effort presented in this paper.

To adaptively develop a PG multigrid hierarchy, representations of LNK must be developed along with the RNK representation. In [23], an eigenvector corresponding to eigenvalue zero (a nontrivial solution to $A\mathbf{x} = \mathbf{0}$) is approximated for a singular M-matrix, with applications to stochastic problems. The left kernel is known a-priori and is perfectly represented on coarse grids, while the accuracy of right kernel is improved with each iteration. In essence, the scheme developed in [23] is a version of the nonsymmetric adaptive setup phase of this paper for a specific problem type with known left kernel. Here, we further generalize αSA to develop LNK as well as RNK.

This paper is organized in the following manner. The rest of this section discusses the importance of singular vectors as near kernel representatives. Section 2 presents the theoretical framework and a two-grid convergence result. Section 3 describes the algorithms and details of their development. Section 4 presents numerical results for two-dimensional convection diffusion problems. Section 5 presents concluding remarks.

1.1. Singular Vectors as Near Kernel. For the SPD setting, it was observed in [17, 18] that the range of interpolation must represent an eigenvector with accuracy

on the same order as the size of the corresponding eigenvalue. This suggests that it is of fundamental importance for the range of interpolation to accurately represent eigenvectors corresponding to very small eigenvalues. How does this generalize to the nonsymmetric setting? The following discussion suggests that the coarse spaces involved in a multigrid hierarchy coupled with a relaxation such as (1.3) should aim to represent singular vectors that correspond to the smallest singular values rather than eigenvectors that correspond to small magnitude eigenvalues. Singular vectors are at least as near-kernel as eigenvectors and are, therefore, more algebraically smooth with respect to (1.3).

Recall the definition of left and right eigenvectors: for any *right eigenvector*, $\mathbf{d}_i$, of A, there exists an *eigenvalue*, $\lambda_i \in \mathbb{C}$, such that

$$A_i\mathbf{d}_i = \lambda_i\mathbf{d}_i, \tag{1.8}$$

and, for any *left eigenvector*, $\mathbf{c}_i$, of A, there exists $\lambda_i \in \mathbb{C}$, such that

$$A^t\mathbf{c}_i = \lambda_i\mathbf{c}_i. \tag{1.9}$$

Define *minimal eigenvalues* as

$$\lambda_j \quad \text{such that} \quad |\lambda_j| = \min_i |\lambda_i|, \tag{1.10}$$

and their corresponding eigenvectors, $\mathbf{c}_j$ and $\mathbf{d}_j$, as *minimal left and right eigenvectors.*

Recall the singular value decomposition:

$$A = U\Sigma V^t, \tag{1.11}$$

where Σ is a non-negative diagonal matrix and U and V are orthogonal matrices. The diagonal entries of Σ, $\sigma_i > 0$, are singular values, while columns of U, $\mathbf{u}_i$, are *left-singular vectors* of A and columns of V, $\mathbf{v}_i$, are *right singular vectors.* Note that

$$A\mathbf{v}_i = \sigma_i\mathbf{u}_i \quad \text{and} \quad A^t\mathbf{u}_i = \sigma_i\mathbf{v}_i. \tag{1.12}$$

Define *minimal singular values* as

$$\sigma_1 \quad \text{such that} \quad \sigma_1 = \min_i \sigma_i, \tag{1.13}$$

and correspondingly, the vectors, $\mathbf{u}_1$ and $\mathbf{v}_1$, as *minimal left and right singular vectors.*

The following version of a standard result, which we present without proof, suggests that minimal singular vectors are at least as near-kernel as the minimal eigenvectors.

THEOREM 1.1. (EIGENVALUE INCLUSION ANNULUS). *Any eigenvalue, λ, of matrix A is located inside an annulus with inner radius σ_1 and outer radius σ_n:*

$$\sigma_1 \leq |\lambda| \leq \sigma_n. \tag{1.14}$$

This theorem emphasizes that a minimal singular vector may be much more near-kernel than a minimal eigenvector:

$$\|A\mathbf{v}_1\| \leq \|A\mathbf{d}_1\| \qquad \text{for} \qquad \|\mathbf{v}_1\| = \|\mathbf{d}_1\| = 1. \tag{1.15}$$

The following example illustrates this result.

EXAMPLE 1.1. Consider matrix A, obtained by applying upwinded finite differences to convection-dominated convection-diffusion in one dimension, scaled by $h = \frac{1}{n-1}$. Matrix A is an $n \times n$ tridiagonal matrix:

$$A = \text{tridiag}[-1, 1, 0] + \frac{\epsilon}{h}\text{tridiag}[-1, 2, -1]. \tag{1.16}$$

For sufficiently weak diffusion ($\epsilon << h$) all eigenvalues of A are $\mathcal{O}(1)$, yet the lowest singular value is $\mathcal{O}(h)$. If $h << 1$, then an error vector that is a minimal right singular vector corresponds to a relatively small residual vector and is algebraically smooth with respect to (1.3). A minimal right eigenvector, however, has a relatively significant residual and is not algebraically smooth.

With this theorem and example in mind, an adaptive multilevel method is designed that concentrates on preserving left and right minimal singular vectors on coarse spaces.

2. Theoretical Framework. Consider the sparse SPD matrices A^tA and AA^t with their eigendecompositions:

$$A^tA = V\Sigma^2V^t \quad \text{and} \quad AA^t = U\Sigma^2U^t, \tag{2.1}$$

where U and V are the left and right singular vector bases from the singular value decomposition (1.11). Because the matrices involved are sparse and SPD, the obvious approaches arise: form a method to solve either the normal equations ($A^tA\mathbf{x} = A^t\mathbf{b}$) or normal residual equations ($AA^t\mathbf{y} = \mathbf{b}$ with $\mathbf{x} = A^t\mathbf{y}$). There are two caveats to these approaches. First, the complexity of the problem matrices has been significantly increased, especially for problems coming from PDEs with high spatial dimension. Second, and of more significance, the singular values in these operators have been squared. Essentially, approximation properties (1.6) and (1.7) are more difficult to attain with respect to A^tA or AA^t, even if the complexity issue could be tolerated or fixed.

Instead, we consider methods for SPD matrices $\sqrt{A^tA}$ or $\sqrt{AA^t}$ because they have the same singular value distribution as A. Define the orthogonal matrix

$$Q := VU^t. \tag{2.2}$$

Like A, matrix Q^t maps the ith right singular vector onto the ith left singular vector, but without scaling by the corresponding singular value:

$$Q^t\mathbf{v}_i = \mathbf{u}_i, \quad \text{and} \quad Q\mathbf{u}_i = \mathbf{v}_i. \tag{2.3}$$

SPD matrices are given by

$$\sqrt{A^tA} := V\Sigma V^t = QA \quad \text{and} \quad \sqrt{AA^t} := U\Sigma U^t = AQ. \tag{2.4}$$

Due to symmetry,

$$QA = A^tQ^t \quad \text{and} \quad AQ = Q^tA^t. \tag{2.5}$$

Matrix Q is used to rewrite the original system, $A\mathbf{x} = \mathbf{b}$, as two different symmetric systems:

$$QA\mathbf{x} = Q\mathbf{b} \tag{2.6}$$

or

$$AQ\mathbf{y} = \mathbf{b} \quad \text{for} \quad \mathbf{x} = Q\mathbf{y}. \tag{2.7}$$

QA and AQ may be full, so it is important to note that we use neither of these operators directly in the algorithm, but only to guide the theory and algorithm development.

Now, consider applying a smoothed aggregation, two-level, coarse-grid correction to either SPD linear system, (2.6) or (2.7). To satisfy either symmetric approximation property, (1.6) or (1.7), minimal eigenvectors should be well-represented by the coarse grids. The minimal eigenvectors of QA are the minimal right singular vectors of A and minimal eigenvectors of AQ are the minimal left singular vectors of A. For this reason, the SA framework is employed to form interpolation operators based on approximations to the minimal left and right singular vectors.

ASSUMPTION 2.1. We assume that the minimal left and right singular vectors of A, $\mathbf{u}_1$ and $\mathbf{v}_1$, are either available or well-approximated using an efficient iterative method. For many problems of interest, adequate methods are formed using the vector of all ones to approximate both the left and right kernel components (LRNK). Additionally, efficient methods are formed from adaptively developed LRNK. See the results in Section 4.

We next introduce some notation. Let $n_f = n$ be the number of fine-level degrees of freedom, n_c the number of coarse-level degrees of freedom, $\mathbf{1}_f$ the vector of ones with length n_f, and $\mathbf{1}_c$ the vector of ones with length n_c.

As in the standard SA framework, formation of the coarse-grid employs an $n_f \times n_c$ *aggregation matrix*, T, composed of zeros and ones. Each column of T corresponds to an aggregate of the fine-grid degrees of freedom, placing a 1 at each row associated with a member of that aggregate. Thus, each row contains one and only one nonzero term. See Section 3.4 or [28] for more explanation of the structure and computation of T. This is an *unsmoothed partition of unity*. Section 3.4 also discusses a smoothed partition of unity in which the elements of T lie in $[0, 1]$, and for which the property $\mathbf{1}_f = T\mathbf{1}_c$ holds. The following discussion applies to either unsmoothed or smoothed partitions of unity.

To form a two-grid correction for system (2.6), an interpolation operator is defined using T, so that $\mathbf{v}_1$ is preserved in the range:

$$P := \text{diag}(\mathbf{v}_1)T, \qquad \text{so} \qquad \mathbf{v}_1 \in \mathcal{R}(P). \tag{2.8}$$

The coarse-grid correction is given by

$$\mathbf{x} \leftarrow \mathbf{x} + P(P^tQAP)^{-1}P^tQ\mathbf{r}, \tag{2.9}$$

where residual vector is $\mathbf{r} = \mathbf{b} - A\mathbf{x}$, which satisfies the residual equation, $A\mathbf{e} = \mathbf{r}$. Then, the corresponding error-propagation operator is given by

$$\mathbf{e} \leftarrow (I - P(P^tQAP)^{-1}P^tQA)\mathbf{e} =: (I - \Pi_1)\mathbf{e}. \tag{2.10}$$

Operator $(I - \Pi_1)$ is a QA-orthogonal projection that projects onto $\mathcal{N}(P^tQA)$ in a direction in $\mathcal{R}(P)$.

For system (2.7), $\mathbf{u}_1$ is preserved in the range of its respective interpolation operator:

$$R := \text{diag}(\mathbf{u}_1)T, \qquad \text{so} \qquad \mathbf{u}_1 \in \mathcal{R}(R). \tag{2.11}$$

A coarse-grid correction, as an iteration in $\mathbf{x}$, is given by

$$\begin{array}{rcll} \mathbf{y} & \leftarrow & \mathbf{y} + R(R^tAQR)^{-1}R^t(\mathbf{b} - AQ\mathbf{y}), & \\ Q\mathbf{y} & \leftarrow & Q\mathbf{y} + QR(R^tAQR)^{-1}R^t(\mathbf{b} - AQ\mathbf{y}), & (\mathbf{x} = Q\mathbf{y}), \\ \mathbf{x} & \leftarrow & \mathbf{x} + QR(R^tAQR)^{-1}R^t\mathbf{r}. & \end{array} \tag{2.12}$$

The error propagation operator of this iteration is

$$\mathbf{e} \leftarrow Q(I - R(R^tAQR)^{-1}R^tAQ)Q^t\mathbf{e} =: (I - \Pi_2)\mathbf{e}. \tag{2.13}$$

Using orthogonality ($Q^tQ = QQ^t = I$), this projection is rewritten as

$$\begin{array}{rcl} (I - \Pi_2) & = & Q(I - R(R^tAQR)^{-1}R^tAQ)Q^t \\ & = & I - QR(R^tQ^tQAQR)^{-1}R^tQ^tQA \\ & = & I - [QR]([QR]^tQA[QR])^{-1}[QR]^tQA. \end{array} \tag{2.14}$$

Operator $(I - \Pi_2)$ is also a QA-orthogonal projection, but one that projects onto $\mathcal{N}(R^tA)$ in a direction in $\mathcal{R}(QR)$.

REMARK 2.2. Projections $(I - \Pi_1)$ and $(I - \Pi_2)$ are both orthogonal in the QA inner product, and this inner product is used for the convergence results in this paper. Unfortunately, the QA-norm of the error components is not computable for problems of practical size.

Again, neither projection leads to an acceptable two-grid method, because it involves full, $n_f \times n_f$ matrices that would yield a method of extremely high complexity. Instead, an oblique projection that involves sparse matrices is used to approximate these orthogonal projections.

2.1. Approximating $(I-\Pi_1)$ and $(I-\Pi_2)$. In (2.10), matrix Q^tP is full, so we replace both occurrences by R to give a two-grid correction of standard complexity:

$$Q^tP = Q^t\text{diag}(\mathbf{v}_1)T \quad \longleftarrow \quad \text{diag}(Q^t\mathbf{v}_1)T = \text{diag}(\mathbf{u}_1)T = R. \tag{2.15}$$

Note that $Q^tP\mathbf{1}_c = R\mathbf{1}_c = \mathbf{u}_1$. The action of these two operators is identical for coarse representation of prototypical algebraically smooth error. Ideally, the action of these two operators will be similar for algebraically smooth error components that are well represented by P.

Similarly, in (2.13), we use P in place of QR:

$$QR = Q\text{diag}(\mathbf{u}_1)T \quad \longleftarrow \quad \text{diag}(Q\mathbf{u}_1)T = \text{diag}(\mathbf{v}_1)T = P. \tag{2.16}$$

These replacements allow us to approximate both $(I - \Pi_1)$ and $(I - \Pi_2)$ with a projection involving sparse intergrid transfer operators and a non-variational coarse-grid,

$$(I - \Pi_a) := (I - P(R^tAP)^{-1}R^tA), \tag{2.17}$$

corresponding to the PG coarse-grid correction

$$\mathbf{x} \longleftarrow \mathbf{x} + P(R^tAP)^{-1}R^t\mathbf{r}. \tag{2.18}$$

Operator $(I - \Pi_a)$ is an oblique projection (not orthogonal with respect to any obvious inner product) that projects onto $\mathcal{N}(R^tA)$ in a direction from $\mathcal{R}(P)$. A summary of the spaces involved is given in Table 2.1 and a simple cartoon is given

	$(I-\Pi_a)$	$(I-\Pi_1)$	$(I-\Pi_2)$
Orthogonality	none	$\langle \cdot, \cdot \rangle_{QA}$	$\langle \cdot, \cdot \rangle_{QA}$
Range	$\mathcal{N}(R^t A)$	$\mathcal{N}(P^t Q A)$	$\mathcal{N}(R^t A)$
Nullspace	$\mathcal{R}(P)$	$\mathcal{R}(P)$	$\mathcal{R}(QR)$

TABLE 2.1
Various projections and the spaces involved

in Figure 2.1. The reader is warned that the low dimensionality of the figure creates many misleading over-simplifications. For one, the correction $\mathbf{e}_a - \mathbf{e}$ is in the same space as correction $\mathbf{e}_1 - \mathbf{e}$, namely, $\mathcal{R}(P)$. However, these two corrections are not necessarily in the same direction, as they appear in the cartoon.

REMARK 2.3. When A is SPD, $Q = I$, $\mathbf{u}_1 = \mathbf{v}_1$, and $R = P$. All three projections are equal, $(I - \Pi_1) = (I - \Pi_2) = (I - \Pi_a)$, and represent Galerkin coarsening.

For our purposes, we consider $(I - \Pi_a)$ to be a good approximation to either of the other two projections if the effect on algebraically smooth error is comparable. We present theoretical results to this end in the next section.

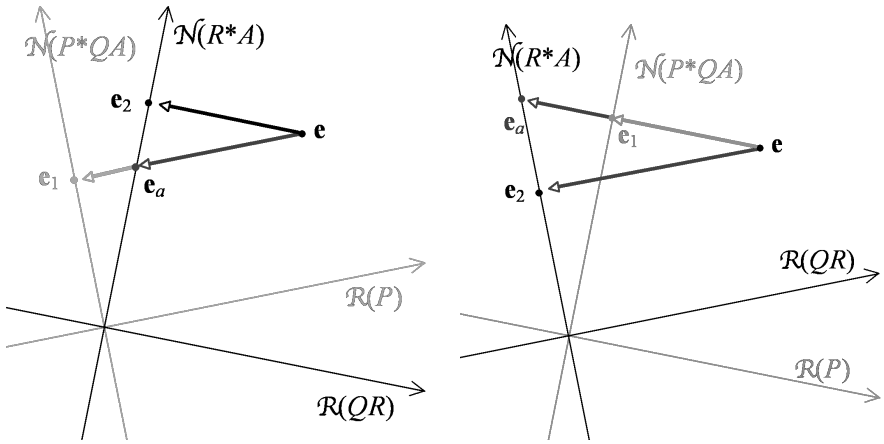


FIG. 2.1. *Two-dimensional cartoons of projections Π_1, Π_2, and Π_a in the QA inner product. The outcome of the projections are labeled as $\mathbf{e}_1 := (I-\Pi_1)\mathbf{e}$, $\mathbf{e}_2 := (I-\Pi_2)\mathbf{e}$, and $\mathbf{e}_a := (I-\Pi_a)\mathbf{e}$. The figure on the left represents a possibility where $\mathbf{e}_a$ is smaller than $\mathbf{e}_2$ in the QA norm, while the figure on the right shows a possibility where $\mathbf{e}_a$ is bigger than $\mathbf{e}_2$. Note that $\mathbf{e}_a$ is always greater than (or possibly equal to) $\mathbf{e}_1$ in the QA norm.*

2.2. Proof of Two-Grid Convergence. This section presents some preliminary two-level convergence theory for non-symmetric smoothed aggregation multigrid. Here, $(I - \Pi_1)$ and $(I - \Pi_2)$ are used as theoretical tools, and $(I - \Pi_a)$ is the actual projection used. The construction in the previous section gives the following useful relationships.

LEMMA 2.1. (PROJECTION IDENTITIES).

$$\begin{array}{rclcrcl} \Pi_1\Pi_a & = & \Pi_a, & & \Pi_a\Pi_1 & = & \Pi_1, \\ \Pi_2\Pi_a & = & \Pi_2, & & \Pi_a\Pi_2 & = & \Pi_a, \end{array} \tag{2.19}$$

Proof. These identities follow from definitions (2.10), (2.13), and (2.17).□

The strong approximation property (SAP) is generalized within this nonsymmetric framework.

ASSUMPTION 2.1. (NONSYMMETRIC STRONG APPROXIMATION PROPERTY): P satisfies the *strong approximation property* with constant K_s if, for any $\mathbf{e}$ on the fine grid, there exists an $\mathbf{e}_c$ on the coarse grid such that

$$\|\mathbf{e} - P\mathbf{e}_c\|_{QA}^2 \leq \frac{K_s}{\|QA\|} \langle QA\mathbf{e},\, QA\mathbf{e}\rangle . \tag{2.20}$$

Note, due to the QA-orthogonality of Π_1 and $Q^tQ = I$, that (2.20) is equivalent to

$$\|(I - \Pi_1)\mathbf{e}\|_{QA}^2 \leq \frac{K_s}{\|A\|} \langle A\mathbf{e},\, A\mathbf{e}\rangle \tag{2.21}$$

for all $\mathbf{e}$ on the fine grid.

Because projection $(I - \Pi_a)$ is oblique, it is necessary to extend the SAP to this specific projection. The following assumption allows us to do so.

ASSUMPTION 2.2. (Π_a STABILITY):

$$\|\Pi_a\|_{QA} < C, \tag{2.22}$$

independent of mesh spacing.

REMARK 2.4. It is not entirely clear how to enforce Assumption 2.2 within a discretization process and respective choice of intergrid transfer operators. However, we have observed Assumption 2.2 for small C (less than 2) for many small convection diffusion problems, using the discretization techniques and grid-transfer operators from this paper. See the Appendix in [21] for more details.

LEMMA 2.2. *If Assumptions 2.1 and 2.2 hold, then*

$$\langle (QA)(I - \Pi_a)\mathbf{e},\, (I - \Pi_a)\mathbf{e}\rangle \leq \frac{C^2 K_s}{\|A\|} \langle A\mathbf{e},\, A\mathbf{e}\rangle , \quad \forall \mathbf{e}. \tag{2.23}$$

Proof. From Lemma 2.1, we have

$$(I - \Pi_a) = (I - \Pi_a)(I - \Pi_1). \tag{2.24}$$

Together with the fact that a projection has the same norm as its complement, this gives

$$\begin{aligned}
\langle (QA)(I - \Pi_a)\mathbf{v},\, (I - \Pi_a)\mathbf{v}\rangle &= \langle (QA)(I - \Pi_a)(I - \Pi_1)\mathbf{v},\, (I - \Pi_a)(I - \Pi_1)\mathbf{v}\rangle \\
&\leq \|I - \Pi_a\|_{QA}^2 \langle (QA)(I - \Pi_1)\mathbf{v},\, (I - \Pi_1)\mathbf{v}\rangle \\
&\leq \frac{C^2 K_s}{\|A\|} \langle (QA)\mathbf{v},\, (QA)\mathbf{v}\rangle .
\end{aligned}$$

This yields the result. □

For the following convergence result, define G to be the error propagation operator of ν iterations of Richardson for the normal equations (see Sections 3.4 for the actual iteration used in our numerical tests):

$$G := \left(I - \frac{1}{\|A\|^2} A^t A\right)^\nu. \tag{2.25}$$

THEOREM 2.3. (TWO-LEVEL QA-CONVERGENCE). *Under the assumptions of Lemma 2.2,*

$$\|(I - \Pi_a)G\mathbf{e}\|_{QA}^2 \leq \frac{16C^2K_s}{25\sqrt{4\nu+1}}\|\mathbf{e}\|_{QA}^2, \tag{2.26}$$

where K_s is the constant from the approximation property.

Proof. Applying Lemma 2.2 to $G\mathbf{e}$ yields

$$\|(I - \Pi_a)G\mathbf{e}\|_{QA}^2 = \langle (QA)(I - \Pi_a)G\mathbf{e}, (I - \Pi_a)G\mathbf{e}\rangle \tag{2.27}$$

$$\leq \frac{C^2K_s}{\|A\|} \langle (QA)G\mathbf{e}, (QA)G\mathbf{e}\rangle \tag{2.28}$$

$$= \frac{C^2K_s}{\|A\|}\|(QA)^{1/2}G\mathbf{e}\|_{QA}^2. \tag{2.29}$$

Decomposing the error in the eigenbasis of QA, $\mathbf{e} = \sum_{j=1}^n \beta_j \mathbf{v}_j$, where $(QA)\mathbf{v}_j = \sigma_j \mathbf{v}_j$, we obtain

$$\|(QA)^{1/2}G\mathbf{e}\|_{QA}^2 = \left\| \sum_{j=1}^n \sigma_j^{1/2} \left(1 - \frac{\sigma_j^2}{\|A\|^2}\right)^\nu \beta_j \mathbf{v}_j \right\|_{QA}^2 \leq s\|\mathbf{e}\|_{QA}^2 \tag{2.30}$$

where

$$s = \sup_{\sigma \in [0, \|A\|^2]} \sigma \left(1 - \frac{\sigma^2}{\|A\|^2}\right)^{2\nu}. \tag{2.31}$$

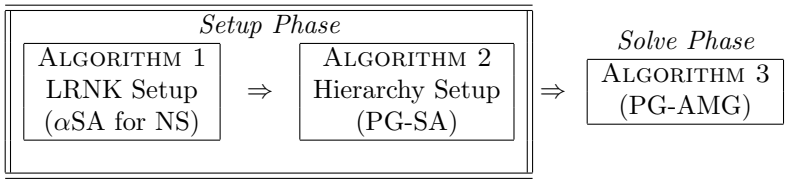
The sup occurs at $\hat{\sigma} = \|A\|/\sqrt{4\nu+1}$, which yields

$$s = \frac{\|A\|}{\sqrt{4\nu+1}} \left(\frac{4\nu}{4\nu+1}\right)^{2\nu} \leq \frac{16\|A\|}{25\sqrt{4\nu+1}} \qquad \text{for} \qquad \nu \geq 1. \tag{2.32}$$

This yields the result. □

The main implication of this theorem is that $\mathcal{O}(K_s^2)$ relaxation steps guarantee convergence of the two-level method. A similar result for a symmetric problem and relaxation $(I - \frac{1}{\|A\|}A)^\nu$ implies that $\mathcal{O}(K_s)$ steps guarantee two-level convergence. This observation is reflected in the multigrid tests on convection-diffusion problems in Section 4, where more relaxation is required for highly nonsymmetric problems than would be for symmetric problems.

FIG. 3.1. *Primary kernel development for nonsymmetric problems.*



ALGORITHM 1 (αSA INITIALIZATION FOR NONSYMMETRIC PROBLEMS).

INPUT: Level l problem matrix A_l, pre- and post- relaxation counts for the setup phase, μ_1 and μ_2, and initial guesses for $\tilde{\mathbf{u}}_l$ and $\tilde{\mathbf{v}}_l$.

OUTPUT: LNK and RNK primary representations, $\tilde{\mathbf{u}}_l$ and $\tilde{\mathbf{v}}_l$.

FUNCTION: $[\tilde{\mathbf{u}}_l, \tilde{\mathbf{v}}_l] = \text{LRNKSetup}_l(\mu_1, \mu_2, \tilde{\mathbf{u}}_l, \tilde{\mathbf{v}}_1)$

1. Pre-relax equations $A_l^t \tilde{\mathbf{u}}_l = \mathbf{0}$ and $A_l \tilde{\mathbf{v}}_l = \mathbf{0}$ μ_1 times with a stationary relaxation method (see section 3.2).
2. If no aggregation is available, build an aggregation $\{\mathcal{A}_j^l\}_{j=1}^{J_l}$ as in Section 3.1.
3. Build intergrid transfer operators, R_{l+1}^l and P_{l+1}^l (see Section 3.4), coarse-grid LRNK, $\tilde{\mathbf{u}}_{l+1}$ and $\tilde{\mathbf{v}}_{l+1}$ (see Section 3.5), and coarse-grid problem matrix $A_{l+1} = (R_{l+1}^l)^t A_l P_{l+1}^l$.
4. If n_{l+1} is small enough to solve directly, then set $L = l + 1$ and move to Step 6. Otherwise, set

$$[\tilde{\mathbf{u}}_{l+1}, \tilde{\mathbf{v}}_{l+1}] = \text{LRNKSetup}_{l+1}(\mu_1, \mu_2, , \tilde{\mathbf{u}}_{l+1}, \tilde{\mathbf{v}}_{l+1}). \tag{3.1}$$

5. Interpolate, $\tilde{\mathbf{u}}_l = R_{l+1}^l \tilde{\mathbf{u}}_{l+1}$ and $\tilde{\mathbf{v}}_l = P_{l+1}^l \tilde{\mathbf{v}}_{l+1}$.
6. Post-relax equations $A_l^t \tilde{\mathbf{u}}_l = \mathbf{0}$ and $A_l \tilde{\mathbf{v}}_l = \mathbf{0}$ μ_2 times with a stationary relaxation method (see section 3.2).

3. αSA for Nonsymmetric Problems. This section describes the αSA framework and its extension to nonsymmetric problems. Three algorithms are presented with a brief explanation of their use. The discussion of the algorithms includes several references to further parts of this section, where motivation and details of the algorithmic components are discussed without distracting from the purpose of these algorithms.

Like typical algebraic multigrid algorithms, αSA has a *setup phase*, where relevant information is extracted from the problem matrix, A, to build a solver, and a *solve phase*, where the solver is employed to improve approximate solutions in an iterative fashion. Here, the setup phase is composed of two stages, the *left and right near-kernel (LRNK) setup* (called *initialization setup phase* in [9]) and the *hierarchy setup* (called *standard SA setup phase* in [9]). First, assuming no LRNK approximations are available, a multilevel technique is used to develop an LRNK representation. The technique used here is similar to the initialization setup phase described in [9], extended to nonsymmetric problems. Second, after an LRNK representation is obtained, a nonsymmetric version of the standard setup phase is employed to build a PG *solver hierarchy* consisting of intergrid transfer operators and problem matrices

ALGORITHM 2 (PG-SA HIERARCHY SETUP).

INPUT: LNK and RNK primary representations, $\tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{v}}_1$, as prescribed in Section 3.2.

OUTPUT: PG multigrid solver hierarchy $\{A_l,\, R_l,\, P_l\}_{l=1}^{L}$.

FUNCTION: HierarchySetup(L)

1. for $l = 1, ..., (L-1)$ do steps (a)-(c)
   (a) If no aggregation is available, build $\{\mathcal{A}_j^l\}_{j=1}^{J_l}$ as in Section 3.1.
   (b) Build intergrid transfer operators, R_{l+1}^l and P_{l+1}^l (see Section 3.4), coarse-grid LRNK, $\tilde{\mathbf{u}}_{l+1}$ and $\tilde{\mathbf{v}}_{l+1}$ (see Section 3.5).
   (c) Build coarse-grid matrix, $A_{l+1} = (R_{l+1}^l)^t A_l P_{l+1}^l$.

ALGORITHM 3 (PG-AMG).

INPUT: PG multigrid solver hierarchy $\{A_l,\, R_l,\, P_l\}_{l=1}^{L}$ including algebraic pre- and post- relaxation methods, initial guess, $\mathbf{x}_l$, and right-hand side, $\mathbf{b}_l$.

OUTPUT: updated iterate, $\mathbf{x}_l$.

FUNCTION: $\mathbf{x}_l = \mathrm{AMG}_l(\mathbf{x}_l, \mathbf{b}_l)$

1. Pre-relax ν_1 times with relaxation method from Section 3.3.
2. Set $\mathbf{b}_{l+1} = (R_{l+1}^l)^t(\mathbf{b}_l - A_l\mathbf{x}_l)$.
3. If $l = L$, then set $\tilde{\mathbf{e}}_{l+1} = A_{l+1}^{-1}\mathbf{b}_{l+1}$. Otherwise, set $\tilde{\mathbf{e}}_{l+1} = \mathrm{AMG}_{l+1}(\mathbf{0}_{l+1}, \mathbf{b}_{l+1})$.
4. Correct $\mathbf{x}_l \leftarrow \mathbf{x}_l + P_{l+1}^l\tilde{\mathbf{e}}_{l+1}$.
5. Post-relax ν_2 times with relaxation method from Section 3.3.

on all levels. Finally, the solver hierarchy is applied as a PG-AMG V-cycle to (1.1) to obtain an approximate solution. The use of these algorithms is summarized in Figure 3.1.

In the description of these three algorithms, we use the following definitions and multilevel notation. A *solver hierarchy* is a collection of various objects that exist on L different *levels*, with level $l = 1$ being the *finest* and $l = L$ being the *coarsest*. Each level contains n_l degrees of freedom and *associated problem matrices*, A_l, which are $n_l \times n_l$ sparse matrices. The fine-level matrix is the original problem matrix from (1.1). Additionally, interpolation operators, P_{l+1}^l, are $n_l \times n_{l+1}$ sparse matrices used to move information from level $l+1$ to level l and restriction operators, $(R_{l+1}^l)^t$, are $n_{l+1} \times n_l$ sparse matrices used to move information from level l to level $l+1$. Here, the solver hierarchy includes a nonstationary relaxation method.

We describe the setup phase that develops the solver hierarchy and then describe the solve phase that uses it. If no LRNK vectors are available, we must first develop them. Algorithm 1 gives the process for automatically generating $\tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{v}}_1$. This is a natural nonsymmetric PG generalization to the initialization setup phase for symmetric matrices from [9]. First, random initial guesses are made for $\tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{v}}_1$. We then use a stationary relaxation technique that smoothes with respect to singular value decomposition (see Section 3.2) on $A_1^t\tilde{\mathbf{u}}_1 = \mathbf{0}$ to develop LNK and $A_1\tilde{\mathbf{v}}_1 = \mathbf{0}$ to develop RNK. An aggregation is built using the technique from Section 3.1. Then, temporary intergrid transfer operators, coarse-grid LRNK representations (see Section 3.4), and coarse problem matrices are all built. The stationary relaxation technique is

used on the coarser level, and the whole process is repeated until the next-to-coarsest level is reached. Stationary post-relaxation is then performed on coarse LRNK, which are brought to the finer level by interpolation. The post-relaxation and interpolation process is repeated until the LRNK is back to the finest level, which is post-relaxed and output as the LRNK representation.

When adequate LRNK vectors are available, we use Algorithm 2 to build the solver hierarchy. This is a PG version of the SA framework [27], as used in [13, 20]. If no aggregation is available, aggregation is built using the technique from Section 3.1. Then, SA is used to build intergrid transfer operators, coarse-grid LRNK representations (see Section 3.5), and coarse problem matrices. This process is repeated until the level L problem matrix is built, giving a full solver hierarchy.

Once a solver hierarchy is built, we use Algorithm 3 to solve (1.1). This is a PG version of an AMG V-cycle iteration. First, nonstationary pre-relaxation from Section 3.3 is applied to the iterate, and then the residual is computed and restricted to the coarse grid . The pre-relaxation and restriction process repeats until the coarsest level is reached, where the error equation is exactly solved. The error is then interpolated and added to the iterate as a correction, and the iterate is post-relaxed with the nonstationary method from Section 3.3. The interpolation, correction, and post-relaxation is repeated until the finest level is reached, completing the V-cycle. The whole V-cycle process is repeated until the iterate has adequately converged.

REMARK 3.1 When the performance of the solver hierarchy is not adequate for a symmetric problem, additional stages could be applied within the setup phase that test and modify the solver hierarchy to improve performance. Specifically, additional *secondary kernel* vectors may be included in the kernel representation, and intergrid transfer operators are formed locally to have good approximation properties for this set of vectors. For example, see the description of the *general setup phase* in [9]. Such a process is of considerable research interest in the nonsymmetric context, because it applies to problems coming from nonsymmetric systems PDEs. However, the development of secondary kernel for nonsymmetric problems will be addressed in future research and not further discussed here.

The rest of this section discusses the details of the various components of these three algorithms. In the following subsections, we abandon the multilevel notation and use a two-level notation that essentially applies to any two adjacent levels in the solver hierarchy. Level l is called the fine grid and level $l+1$ the coarse grid. Symbols with subscript c represent properties or objects on the coarse grid. Symbols without a subscript or with subscript f describe fine-grid objects. For example, n_f is the number of degrees of freedom on the fine grid and n_c is the number on the coarse grid.

3.1. Absolute-Symmetrized Aggregation. An *aggregation* is a list of sets, $\{\mathcal{A}_j\}_{j=1}^{n_c}$, that form a disjoint covering of the fine-grid degrees of freedom, $\mathcal{A}_j \cap \mathcal{A}_i = \emptyset$ and $\bigcup_{j=1}^{n_c} \mathcal{A}_j = \{1, ..., n_f\}$. Each $\mathcal{A}_j$ is called an *aggregate* and is a local group in the sense that any two degrees of freedom in the set are close within the graph of matrix A. Here, we represent the aggregation as an $n_f \times n_c$ matrix,

$$T_{ij} = \begin{cases} 1 & i \in \mathcal{A}_j \\ 0 & i \notin \mathcal{A}_j \end{cases}, \tag{3.2}$$

whose columns form a *partition of unity*, having properties $\mathbf{1}_f = T\mathbf{1}_c$ and $T \geq 0$.

Our aggregates are based on a measure of connection strength within the graph of matrix A that we call *absolute-symmterized, distance-one connection.* Points i and

j are considered to be strongly connected if

$$|b_{ij}| \geq \zeta \sqrt{b_{ii} b_{jj}} \qquad \text{with} \qquad B = \frac{1}{2}\left(|A| + |A^t|\right), \tag{3.3}$$

where $\zeta \in [0, 1)$ is chosen to filter out weak connections. The partitioning algorithm employed ensures that each aggregate contains at least all points strongly connected to a central *seed* point. See [28] for details of the aggregation algorithm.

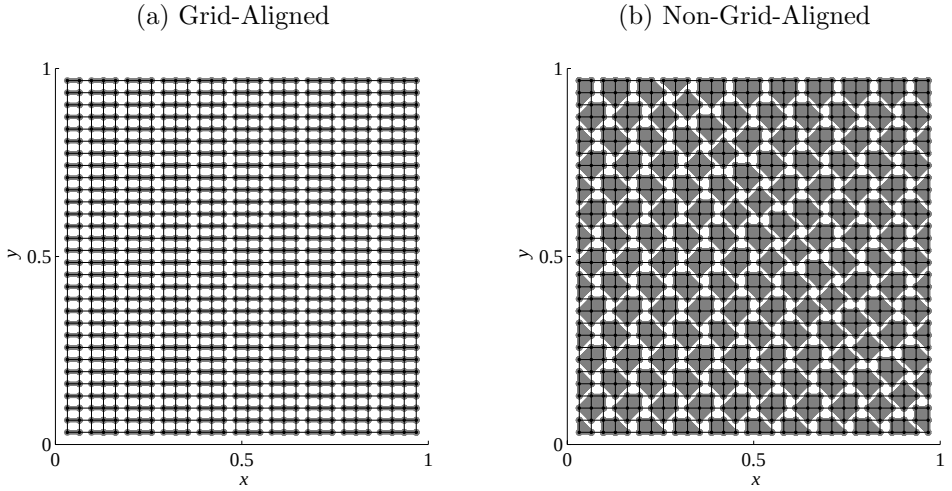


FIG. 3.2. *First level aggregations for convection dominated versions of Example 4.1 of Section 4 on a* 30×30 *grid. Black dots represent fine-level nodes. Each gray shape enclosing dots is an aggregate, representing a coarse node. For the aggregation technique in this paper, semi-coarsening occurs for strong, grid-aligned coupling but not for strong, non-grid-aligned coupling.*

Note that this aggregation scheme tends to result in semi-coarsening for problems exhibiting strong numerical anisotropies as displayed in Figures 3.2 and 4.3.

3.2. LRNK Representation. Algorithm 2 requires LRNK approximations, $\tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{v}}_1$. These could simply be the constant vector, as was used in original SA methods [25] and in some recent nonsymmetric versions [20]. More generally, however, this should be done in an adaptive way to accurately represent LRNK.

In Section 4, numerical results are presented for three types of NK approximations, which are described in the next several paragraphs.

(K0) Constant Near-Kernel. For discretized differential operators, constant vectors often provide sufficiently accurate approximations to LRNK (the row sum is equal to zero for all equations that do not involve boundary nodes). For this reason, constant vectors are a common choice for LRNK approximations:

$$\tilde{\mathbf{u}} \longleftarrow \mathbf{1}_f \qquad \text{and} \qquad \tilde{\mathbf{v}} \longleftarrow \mathbf{1}_f \tag{3.4}$$

Note that the solvers formed using these LRNK approximations skip the αSA NK setup given in Algorithm 1. Such an approach uses a priori knowledge regarding the nature of A, namely that it is a discretized differential operator. For problems where constants are not adequate LRNK approximations, using constants for LRNK may result in poor convergence of the resulting multigrid hierarchies.

(K1) Adaptive Near-Kernel. Algorithm 1 is employed to develop LRNK:

$$[\check{\mathbf{u}}, \check{\mathbf{v}}] \longleftarrow \text{LRNKSetup}_1(\mu_1, \mu_2, \mathbf{1}_f, \mathbf{1}_f). \tag{3.5}$$

The relaxation techniques used in this algorithm are stationary iterations that smooth with respect to the singular value decomposition. Specifically, we use Richardson iteration on the normal equations for $A\mathbf{x} = \mathbf{0}$ and Richardson iteration on the normal-residual equations for $A^t\mathbf{y} = \mathbf{0}$.

For the normal equations and zero right-hand side, Richardson iteration is

$$\mathbf{x} \longleftarrow (I - \alpha A^t A)\mathbf{x}, \tag{3.6}$$

where α is chosen for good smoothing properties. For the normal-residual equations with zero right-hand side, Richardson iteration is

$$\mathbf{y} \longleftarrow (I - \alpha A A^t)\mathbf{y}, \tag{3.7}$$

Parameter α is the same for both iterations:

$$\alpha = \frac{8}{5\|A\|^2}, \tag{3.8}$$

which best attenuates the singular vectors that correspond to singular values in the interval $[\|A\|/2, \|A\|]$, for a general distribution of singular values.

(K2) Singular Vector Near-Kernel. Finally, consider using minimal left and right singular vectors, $\mathbf{u}$ and $\mathbf{v}$, as LRNK approximations. Assume that *accurate* approximations to singular vectors can be computed efficiently. We use the term "accurate" to mean that the solver built with the near-kernel approximations is optimal. The work in [10] suggests that this assumption is reasonable.

Methods based on minimal singular vectors are investigated to assert that approximate minimal singular vectors are LRNK representations of interest. We compute these with the `matlab` function `svds` for the small test problems in Section 4.

REMARK 3.1. As it is currently implemented, using left and right minimal singular vectors as LRNK as in (K2) is currently not efficient, unless a discretization package has provided them. A sparse, multilevel singular-value solver (similar to the eigensolvers in [10, 14]) should be implemented when these vectors are not readily available. Nevertheless, tests involving (K2) are presented to assert that minimal singular vectors used as LRNK are sufficient to form acceptable multigrid hierarchies and are used for comparison with the computationally reasonable methods, (K0) and (K1).

3.3. USYMQR: Solve Phase Relaxation. We use a different relaxation in the near-kernel setup (Algorithm 1) than in the solve phase (Algorithm 3). The solve phase uses a small number (2 to 10) of iterations of a nonstationary Krylov-like method, USYMQR [22], based on spaces of adjoint powers of A in a MINRES-type algorithm. We expect the nonstationary USYMQR to work better as a smoother in the final solver, and rely on stationary Richardson for the normal equations when developing LRNK. First, for a current approximation, we rewrite (1.1) in terms of the error, $A\mathbf{e} = \mathbf{r}$, and apply USYMQR to this equation. We use $\mathbf{r}$ for both *generating vectors*, as suggested (see [22] for details). This form of USYMQR chooses a vector of minimal residual in the affine space

$$\mathcal{S}_k := \mathbf{e} + \text{span}\big\{A^tA\mathbf{e}, ..., (A^tA)^k\mathbf{e}, A\mathbf{e}, A^tAA\mathbf{e}, ..., (A^tA)^kA\mathbf{e}\big\}, \tag{3.9}$$

which is rewritten as

$$\mathbf{e} + \text{span}\{A^tA\mathbf{e}, ..., (A^tA)^k\mathbf{e}\} \oplus \text{span}\{A\mathbf{e}, A^tAA\mathbf{e}, ...(A^tA)^kA\mathbf{e}\}, \tag{3.10}$$

to emphasize that multiple applications of Richardson on the normal equations produce a vector in $\mathcal{S}_k$:

$$(I - \alpha A^tA)^k\mathbf{e} \quad \in \quad \mathbf{e} + \text{span}\{A^tA\mathbf{e}, ..., (A^tA)^k\mathbf{e}\} \quad \subset \quad \mathcal{S}_k. \tag{3.11}$$

Thus, in terms of residual error, USYMQR gives a better vector than Richardson iteration for the normal equations with the same number of matrix-vector evaluations.

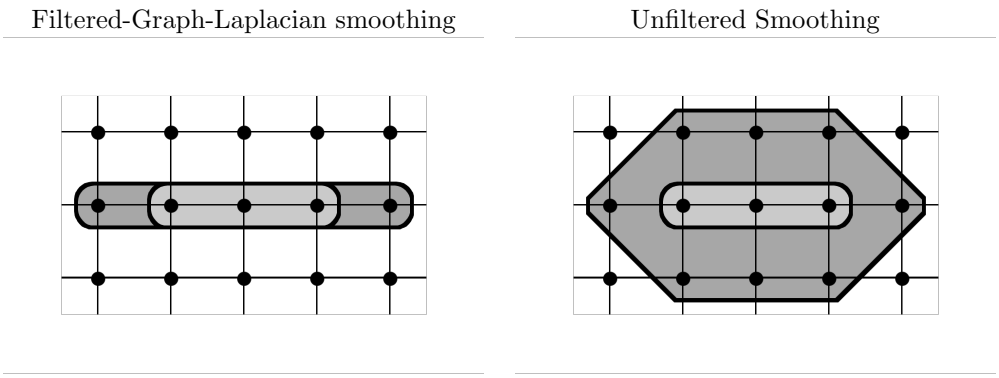


FIG. 3.3. *Interpolation (or restriction) for grid-aligned anisotropies. Black dots represent fine-level nodes, light gray groupings represent a single aggregate, and dark gray groupings represent the support of a single column of interpolation (or a row of restriction). Compare the support of the filtered Graph-Laplacian smoothing (which only spreads in the direction of strong connection) to that of standard, unfiltered smoothing (which spreads weakly in the direction of weak connection). The unfiltered smoothing leads to high complexities, as seen in Figure 3.4*

3.4. Filtered-Graph-Laplacian Operator Smoothing. One-dimensional aggregates are formed in solver hierarchies for problems with strong coupling in grid-aligned directions. Such aggregates can cause original SA to create a solver hierarchy with unacceptable operator complexity, unless special measures are taken to curb complexity growth [28, 8, 12]. This section presents a new approach to matrix filtering that differs from previous matrix filtering approaches. We smooth the partition of unity directly, instead of smoothing a tentative interpolation operator as is done in the original SA framework. Interpolation has the form

$$P \longleftarrow \text{diag}(\tilde{\mathbf{v}})\, S_G\, T, \tag{3.12}$$

and restriction

$$R \longleftarrow \text{diag}(\tilde{\mathbf{u}})\, S_G\, T, \tag{3.13}$$

where S_G is a smoothing operator that preserves the partition-of-unity properties ($\mathbf{1}_f = S_G T \mathbf{1}_c$ and $S_G T \geq 0$). The form of S_G used in this work is

$$S_G = I - \frac{2}{3} D_G^{-1} G, \tag{3.14}$$

where G is a *filtered-graph-Laplacian* of A (defined below) and D_G is the diagonal part of G.

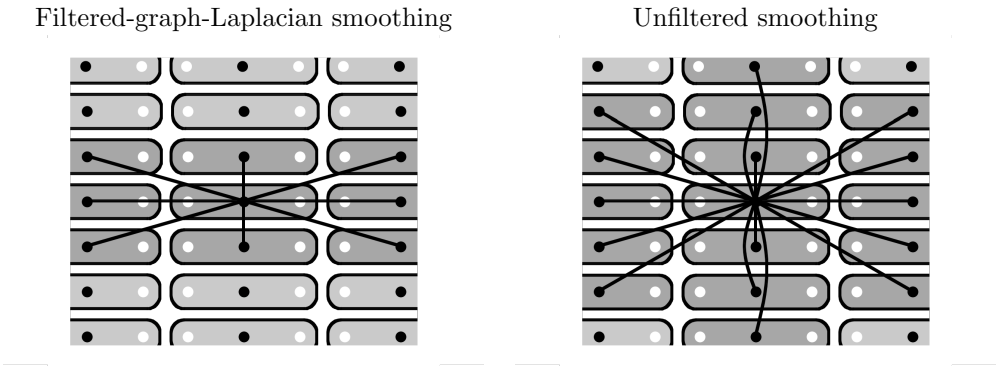


FIG. 3.4. *Coarse-grid stencils for problems with grid-aligned anisotropies. Compare the increase of complexity for the filtered-graph-Laplacian smoothing (5-pt to 9-pt) to that of standard, unfiltered smoothing (5-pt to 17-pt). The operator complexity of the unfiltered case grows unboundedly with the number of levels, L, in the hierarchy.*

Matrix G is defined in terms of the strong connections within the graph of matrix A, as determined by (3.3):

$$G_{ij} = \left\{ \begin{array}{ll} -1 & i \neq j, \quad i \text{ is strongly connected to } j \\ \sum_{k \neq i} G_{ik} & i = j \end{array} \right\} \tag{3.15}$$

This matrix has the important property that $G\mathbf{1}_f = \mathbf{0}_f$, which in turn implies that $\mathbf{1}_f = S_G T \mathbf{1}_c$. Under this construction, $S_G T$ is a partition of unity ($S_G \geq 0$, so $S_G T \geq 0$).

This approach is a novel way to avoid unbounded operator complexities for problems with grid-aligned anisotropies that generalizes nicely to nonsymmetric matrices. The standard filtering techniques of [28] use A_F, a filtered version of the original matrix, to smooth interpolation. The filtering lumps small off-diagonal entries onto other large entries within the same row in a manner that preserves the near kernel property of $\tilde{\mathbf{v}}$. Standard-filtered interpolation is given by

$$(I - \alpha A_F)\text{diag}(\tilde{\mathbf{v}})T. \tag{3.16}$$

We do not use this filtering technique here because the smoothing does not necessarily preserve the minimal singular vector in the range of interpolation. For example, if no filtering was necessary ($A_F = A$), then $A\mathbf{v}_1 = \sigma_1 \mathbf{u}_1$ and

$$(I - \alpha A_F)\text{diag}(\tilde{\mathbf{v}})T\mathbf{1}_c = \mathbf{v}_1 - \alpha\sigma_1 \mathbf{u}_1, \tag{3.17}$$

which mixes the left and right singular vectors unless $\sigma_1 = 0$. In general, the minimal left and right singular vectors are not linearly dependent and the NK representation has been slightly distorted.

3.5. Coarse-Grid LRNK Representation. Coarse representations are set to match the fine LRNK vectors exactly:

$$\tilde{\mathbf{u}} = R\tilde{\mathbf{u}}_c \qquad \text{and} \qquad \tilde{\mathbf{v}} = P\tilde{\mathbf{v}}_c, \tag{3.18}$$

when vectors $\tilde{\mathbf{u}}_c = \tilde{\mathbf{v}}_c = \mathbf{1}_c$. Typically, scaling the columns of R and P by diagonal matrices N_R and N_P is necessary. Intergrid transfer operators are scaled:

$$R \longleftarrow RN_R \qquad \text{and} \qquad P \longleftarrow PN_P, \tag{3.19}$$

while the coarse LRNK representation is scaled accordingly so that (3.18) holds:

$$\tilde{\mathbf{u}}_c \longleftarrow N_R^{-1}\mathbf{1}_c \qquad \text{and} \qquad \tilde{\mathbf{v}}_c \longleftarrow N_P^{-1}\mathbf{1}_c. \tag{3.20}$$

4. Numerical Results. We investigate the performance of αSA on two-dimensional *convection-diffusion* problems with various convection fields. The results in [20] show that classical unsmoothed and smoothed aggregation methods were far from optimal for problems of this type.

The convection-diffusion operator is posed on the unit square with Dirichlet boundaries:

$$\begin{aligned} -\epsilon\Delta u + \underline{b}\cdot\nabla u &= f \quad \text{in} \quad \Omega = (0,1)^2 \\ u &= 0 \quad \text{in} \quad \partial\Omega, \end{aligned} \tag{4.1}$$

where ϵ is a diffusion parameter and $\underline{b}$ is a divergence-free convection field, $\nabla\cdot\underline{b} = 0$. The examples in this section consider three types of convection fields: constant (Example 2), variable (Example 3), and recirculating (Example 4).

Equation (4.1) is discretized with 5-point finite difference stencils with upwinding. The so-called *mesh-Péclet number*, or *grid-Reynolds number*,

$$\gamma = \frac{|\underline{b}|h}{\epsilon}, \tag{4.2}$$

characterizes what quantity of upwinding is necessary for a stable discretization. Locations of the domain for which $\gamma >> 1$ are convection dominated, and a significant amount of upwinding must be applied. The stencil for this discretization is from [19] and is given by

$$\frac{1}{h^2}\begin{bmatrix} & -\epsilon + bh\mu_y & \\ -\epsilon + ah(\mu_x - 1) & -\Sigma & -\epsilon + ah\mu_x \\ & -\epsilon + bh(\mu_y - 1) & \end{bmatrix}, \tag{4.3}$$

where

$$\mu_x = \begin{cases} \epsilon/2ah & \text{if} \quad ah > \epsilon \\ 1 + \epsilon/2ah & \text{if} \quad ah < -\epsilon \\ \frac{1}{2} & \text{if} \quad |ah| \le \epsilon \end{cases} \tag{4.4}$$

and

$$\mu_y = \begin{cases} \epsilon/2bh & \text{if} \quad bh > \epsilon \\ 1 + \epsilon/2bh & \text{if} \quad bh < -\epsilon \\ \frac{1}{2} & \text{if} \quad |bh| \le \epsilon. \end{cases} \tag{4.5}$$

Various tests were performed for different levels of convection with Péclet numbers $\gamma = 10^{-1}$ (diffusion-dominated), $\gamma = 10^{1}$ (mildly convection-dominated), and $\gamma = 10^{3}$ (convection-dominated). Note that two problems with equal γ and different mesh sizes are approximating two different continuous problems with convection fields of different magnitude.

The methods used for Examples 2, 3, and 4 are reported in the next several paragraphs. Three types of LRNK representation are used in each example: **(K0)** constant kernel, **(K1)** αSA kernel from Algorithm 1, or **(K2)** SVD kernel. Then Algorithm 2 is used to create a PG-AMG multigrid hierarchy based on these various

kernel types. Finally, Algorithm 3 is applied to the homogeneous problem, $A\mathbf{x} = \mathbf{0}$, with initial guess of random entries in $[-.5, .5]$.

The aggregation technique used in all tests is symmetrized, strength-of-connection based, distance-one aggregation (as discussed in Section 3.1) with $\zeta = .05$ (see (3.3)). The first-level aggregations are displayed for small, convection-dominated problems in Figure 3.2, for the grid-aligned and non-grid-aligned cases (Example 2), and in Figure 4.3, for variable and recirculating convection fields (Example 3 and 4). Semi-coarsening is achieved in portions of domain with grid-aligned dominant convection, as expected. The coarsening in portions of the domain with non-grid-aligned dominant convection is similar to that of a diffusion-dominated problem.

The intergrid operator smoothing technique used in all tests was filtered-graph-Laplacian smoothing, as discussed in Section 3.4. This technique gave bounded operator complexities below 2 in all cases. See the examples for specific operator complexity bounds.

The relaxation used in Algorithm 1 for the αSA-developed LRNK in each example is Richardson on the normal and normal-residual equations, as discussed in Section 3.1. For most tests, $\mu_1 = \mu_2 = 5$ (number of pre- and post- relaxation steps on each level of the adaptive setup). When more relaxation was used in the adaptive phase, it is reported in the description of the specific example.

In all tests, the relaxation used for the solver was USYMQR [22], as discussed in Section 3.3. It was observed that V(ν_1, 0)-cycles give the best performance in terms of work units per digit accuracy. Various numbers of pre-relaxation sweeps were used, based on the mesh-Péclet number. See the examples for specific values ν_1 used for each problem type.

The performance of each solver formed is reported as work units per digit of accuracy, η, a function of asymptotic convergence factor and operator complexity. *Asymptotic convergence factors* are estimated by taking the geometric average of the residual reduction for the last 5 of 25 V-cycles:

$$\rho \approx \left(\frac{\|\mathbf{r}^{(25)}\|}{\|\mathbf{r}^{(20)}\|} \right)^{1/5}. \tag{4.6}$$

Operator complexities are reported as the sum of non-zeros in the problem matrices on each level, divided by the number of nonzeros in the original problem matrix,

$$\sigma^A = \frac{\sum_{l=1}^{L} \mathrm{nz}(A_l)}{\mathrm{nz}(A_1)}. \tag{4.7}$$

These two values are used to report a measure, η, which is defined below to quantify how much work is necessary for a certain amount of error reduction.

DEFINITION 4.1 (WORK UNITS PER DIGIT OF ACCURACY) A measure of comparison for several different methods with different numbers of relaxation steps, different types of relaxation, and on very different algebraic grids is required. The *time taken* to reach a desired relative residual would be the preferred measure for gauging the success of these methods, however, our high-level `matlab` implementation is not an environment that is reasonable for timing. Instead, we compare all these methods with an estimation of computational cost required to increase the accuracy of an approximate solution by one order of magnitude (*work units per digit of accuracy*):

$$\eta = \sigma^A(\omega\nu_1 + \omega\nu_2 + 1)\frac{\log .1}{\log \rho}, \tag{4.8}$$

where ω is the number of work units per relaxation step and ν_1 and ν_2 are the numbers of pre- and post-relaxation steps, respectively. We set $\omega = 2$ for USYMQR, because each iteration applies both A and A^t. Term $(\omega\nu_1 + \omega\nu_2 + 1)$ is the number of *residual evaluations* required per level for pre-relaxation, coarse-grid correction, and post-relaxation.

As a frame of reference, Table 4.1 lists values of η for various asymptotic convergence factors, Gauss-Seidel relaxation ($\omega = 1$), and V(1,1)-cycles with $\sigma^A = 2.25$, which we considered as a typical operator complexity for classical AMG for Poisson-like problems with two-dimensional finite difference stencils (see Stuben's appendix in [24]).

REMARK 4.2. Note that the measure, η, only considers residual evaluations on each level, or matrix-vector multiplies with A_l. It does not factor the cost of interpolation, restriction, and other computational costs, whereas monitoring the timing would.

ρ	0.05	0.1	0.2	0.3	0.4	0.5
η	(5.19)	(6.75)	(9.67)	(12.91)	(16.96)	(22.42)
ρ	0.6	0.7	0.75	0.8	0.85	0.9
η	(30.43)	(43.58)	(54.03)	(69.65)	(95.63)	(147.52)

TABLE 4.1

WU-per digit accuracy, η, for methods of certain asymptotic convergence factors, ρ, with Gauss-Seidel relaxation, V(1,1) cycles, and operator complexity of 2.25. To be used as a frame-of-reference to gauge how well the methods are performing. Typical AMG convergence factors for Poisson problems are $\rho = 0.1$ [24].

EXAMPLE 4.1. (CONSTANT CONVECTION FIELD) Consider Problem (4.1) with diffusion $\epsilon = 1$ and a constant convection field. Tests were performed with different problem sizes, mesh-Péclet numbers, and angles of convection. The convection field, $\underline{b} = [b_1, b_2]^t$, is of the form,

$$b_1 = \frac{\gamma\epsilon}{h}\cos\theta, \qquad b_2 = \frac{\gamma\epsilon}{h}\sin\theta, \qquad \theta = \{0^o, 22.5^o\}. \tag{4.9}$$

The angles were chosen to show how the method handles the cases of grid-aligned and non-grid-aligned convection: $\theta = 0^o$ is a grid-aligned case and $\theta = 22.5^o$ is a non-grid-aligned case. Due to the use of ordering-independent relaxation methods, these are the only two directions for which we report results. An ordering-dependent relaxation method, such as Gauss-Seidel, would need a larger sample of directions.

Different numbers of pre-relaxation sweeps were used for different orders of convection: tests with $\gamma = .1$ used V(2,0)-cycles; tests with $\gamma = 10$ used V(5,0)-cycles; and tests with $\gamma = 1000$ used V(5,0)-cycles.

For the methods formed with αSA LRNK, asymptotic convergence factors were found to be under 0.26 for $\gamma = .1$ (where V(2,0) cycles were used), under .09 for $\gamma = 10$ problems (with V(5,0) cycles), and under .02 for $\gamma = 1000$ (with V(5,0) cycles).

The operator complexities of the solvers created were under 1.91 for $\gamma = 10^3$ with grid-aligned convection and under 1.37 for all other problems.

Figure 4.1 reports work units per digit of accuracy (as calculated with Formula (4.8)) for various problem sizes, convection magnitudes, and convection directions. For the diffusion-dominated problems, considerable improvement is made by using αSA kernel development, versus using constant LRNK.

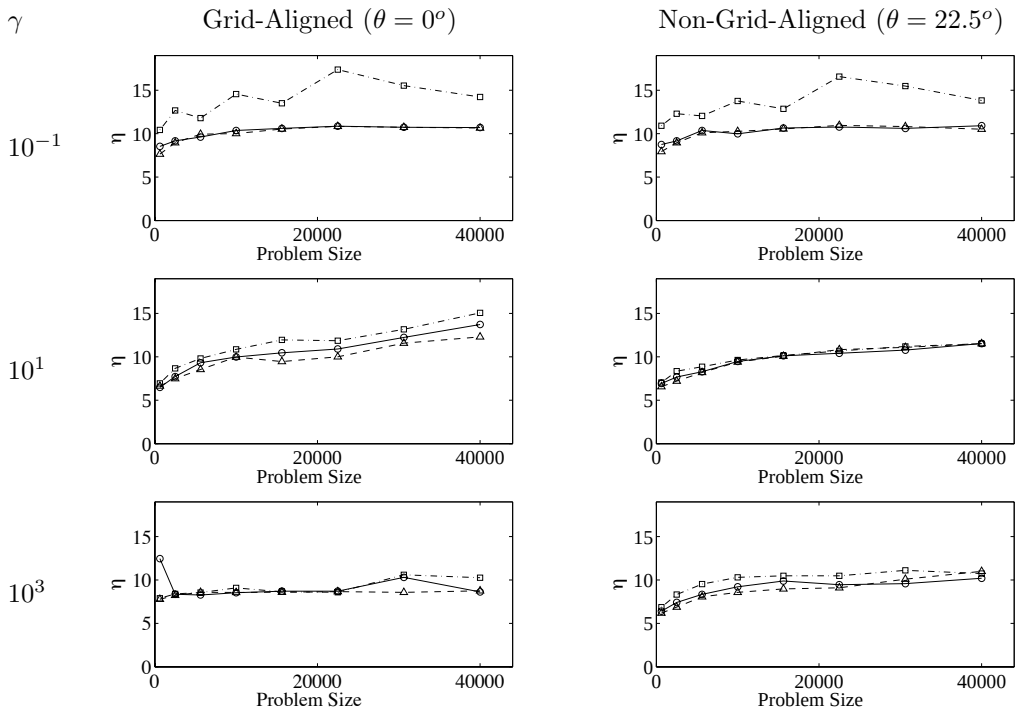


FIG. 4.1. *Work units per digit of accuracy, η, for Example 4.1 across parameters θ, γ, and n.* **(K0)** *dash-dotted lines with square markers stand for solvers formed using constants as LRNK.* **(K1)** *solid lines with circle markers stand for solvers formed using αSA LRNK.* **(K2)** *dashed lines with triangle markers stand for solvers formed using SVD LRNK. The left column of graphs presents results for grid-aligned convection, the right column for non-grid-aligned. The first row presents results for mesh-Péclet number .1, the second 10, and the third 1000. Problem sizes ranged from 625 to 40000. Number of levels used ranged from 3 to 7. Operator complexities were below 1.37 in every case except the convection-dominated grid-aligned case, where the complexity was below 1.91.*

For the problems where αSA was used to approximate LRNK, measure η was less than 14, which is comparable to a typical classical AMG Poisson-like rate of 0.33 (see Table 4.1).

EXAMPLE 4.2. (VARIABLE CONVECTION FIELD) Next, consider Problem (4.1) with $\epsilon = 1$ and variable convection fields. Again, the various tests have different problem sizes and mesh-Péclet numbers of various order. For each problem, the angle of convection changes throughout the domain, giving both grid-aligned and non-grid-aligned cases. The convection field is from [19] and is given by

$$b_1 = \frac{\gamma\epsilon}{h}(2y-1)(1-x^2), \qquad b_2 = \frac{\gamma\epsilon}{h}2xy(y-1). \tag{4.10}$$

Note that $\max_\Omega |\underline{b}| = \gamma\epsilon h^{-1}$. Similar problems are seen in [20] and referred to there as `bent-pipe` problems. See Figure 4.2(a) for an example.

Different numbers of pre-relaxation sweeps were used for different orders of convection: tests with $\gamma = .1$ used V(2,0)-cycles; tests with $\gamma = 10$ used V(5,0)-cycles; and tests with $\gamma = 1000$ used V(7,0)-cycles.

The operator complexities of the solvers created were under 1.35 for $\gamma = 10^{-1}, 10$, and under 1.48 for $\gamma = 10^3$.

Table 4.2 reports asymptotic convergence estimates, work units per digit of accu-

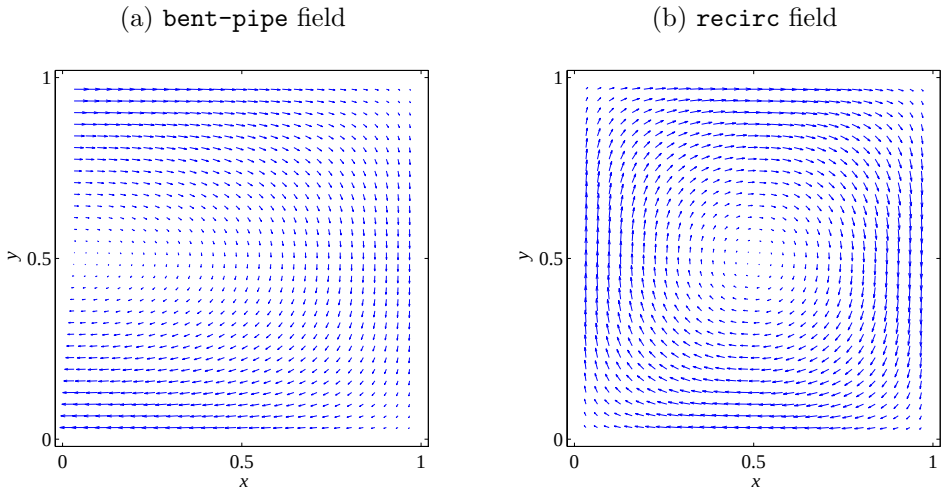


FIG. 4.2. *Convection fields for Examples 4.2 and 4.3 on a* 30×30 *grid.*

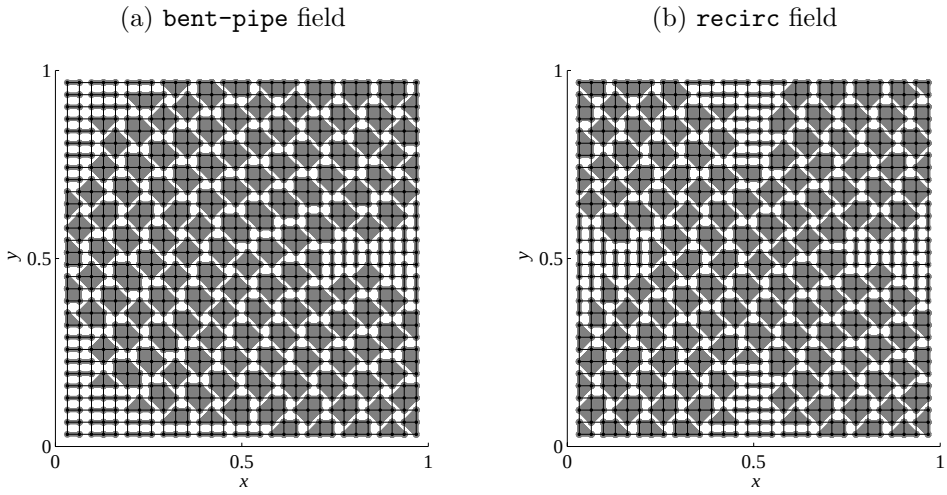


FIG. 4.3. *First level aggregations for Examples 4.2 and 4.3 on a* 30×30 *grid with* $\gamma = 10^3$. *Black dots represent fine-level nodes. Each gray shape enclosing dots is an aggregate, representing a coarse node. For the aggregation technique in this paper, semi-coarsening occurs for areas in the domain with strong, grid-aligned coupling but not for areas with strong, non-grid-aligned coupling.*

racy (as calculated with Formula (4.8)), and number of levels used for various problem sizes, convection magnitudes, and LRNK approximation types.

EXAMPLE 4.3. (RECIRCULATING CONVECTION FIELD) Next, consider Problem (4.1) with $\epsilon = 1$ and variable convection fields. Again, we run tests on many problems with different magnitudes of convection, $\gamma = 10^{-1}, 10^1, 10^3$. The convection field is from [19] and is given by

$$b_1 = \frac{\gamma\epsilon}{h} 4x(x-1)(1-2y), \qquad b_2 = -\frac{\gamma\epsilon}{h} 4y(y-1)(1-2x). \tag{4.11}$$

Again, $\max_\Omega |\underline{b}| = \gamma\epsilon h^{-1}$ and similar problems are seen in [20, 31], referred to there

		Constant LRNK			αSA LRNK			SVD LRNK		
n^2	γ	ρ	η	L	ρ	η	L	ρ	η	L
16^2	10^{-1}	0.194	(9.15)	2	0.115	(6.94)	2	0.129	(7.33)	2
	10^{1}	0.013	(7.58)	2	0.016	(8.00)	2	0.020	(8.48)	2
	10^{3}	0.002	(8.04)	3	0.002	(7.64)	3	0.002	(8.01)	3
32^2	10^{-1}	0.321	(13.37)	3	0.229	(10.29)	3	0.133	(7.53)	3
	10^{1}	0.032	(9.76)	3	0.029	(9.48)	3	0.045	(10.75)	3
	10^{3}	0.012	(11.27)	3	0.009	(10.66)	3	0.009	(10.74)	3
64^2	10^{-1}	0.373	(15.66)	4	0.246	(11.03)	4	0.210	(9.91)	4
	10^{1}	0.059	(12.03)	4	0.032	(9.92)	4	0.061	(12.16)	4
	10^{3}	0.025	(13.78)	4	0.019	(12.77)	4	0.017	(12.40)	4
128^2	10^{-1}	0.330	(13.84)	4	0.278	(12.01)	4	0.238	(10.69)	4
	10^{1}	0.084	(13.68)	4	0.095	(14.38)	4	0.108	(15.24)	4
	10^{3}	0.047	(16.63)	5	0.047	(16.62)	5	0.045	(16.35)	5

TABLE 4.2

Variable convection field results. Operator complexities were under 1.35 for $\gamma = 10^{-1}, 10$, and under 1.48 for $\gamma = 10^3$.

as called `recirc` problems. See Figure 4.2(b) for an example.

Different numbers of pre-relaxation sweeps were used for different orders of convection: tests with $\gamma = .1$ used V(2,0)-cycles; tests with $\gamma = 10$ used V(5,0)-cycles; and tests with $\gamma = 1000$ used V(9,0)-cycles.

The operator complexities of the solvers created were under 1.35 for $\gamma = 10^{-1}, 10$, and under 1.48 for $\gamma = 10^3$.

Table 4.3 reports asymptotic convergence estimates, work units per digit of accuracy (as calculated with Formula (4.8)), and number of levels used for various problem sizes, convection magnitudes, and LRNK approximation types.

		Constant LRNK			αSA LRNK			SVD LRNK		
n^2	γ	ρ	η	L	ρ	η	L	ρ	η	L
16^2	10^{-1}	0.263	(11.52)	3	0.139	(7.80)	3	0.125	(7.41)	3
	10^{1}	0.130	(16.63)	3	0.082	(13.60)	3	0.068	(12.63)	3
	10^{3}	0.028	(16.16)	2	0.029	(16.27)	2	0.021	(14.80)	2
32^2	10^{-1}	0.327	(13.62)	4	0.136	(7.62)	4	0.155	(8.16)	4
	10^{1}	0.243	(23.73)	4	0.143	(17.25)	4	0.128	(16.28)	4
	10^{3}	0.108	(28.00)	3	0.111	(28.37)	3	0.085	(25.34)	3
64^2	10^{-1}	0.384	(16.16)	5	0.211	(9.93)	5	0.196	(9.49)	5
	10^{1}	0.435	(40.83)	5	0.283	(26.96)	5	0.264	(25.52)	5
	10^{3}	0.210	(40.67)	4	0.217	(41.65)	4	0.214	(41.20)	4
128^2	10^{-1}	0.466	(20.11)	5	0.238	(10.70)	5	0.234	(10.56)	5
	10^{1}	0.643	(76.55)	5	0.499	(48.76)	5	0.388	(35.79)	5
	10^{3}	0.392	(68.61)	5	0.397	(69.45)	5	0.376	(65.81)	5

TABLE 4.3

Recirculating convection field results. Operator complexities were under 1.35 for $\gamma = 10^{-1}, 10$, and under 1.48 for $\gamma = 10^3$.

4.1. Comparison with AMG. Next, we show how AMG performs in comparison on various instances of Examples 2, 3, 4. The AMG results are reported in [19]

on 64^2 grids, and we plugged the estimated convergence factors and operator complexities from this book into (4.8) with $\omega = 1$ to get values for η, which are displayed in Table 4.4.

			classical AMG			αSA for NS		
problem	ϵ	γ	ρ	σ^A	η	ρ	σ^A	η
$\theta = 0^o$	10^{-5}	1.54e+03	4e-4	2.30	(2.03)	5e-4	1.89	(8.45)
$\theta = 22.5^o$	10^{-5}	1.54e+03	.005	4.45	(5.80)	.003	1.34	(8.10)
$\theta = 45^o$	10^{-5}	1.54e+03	5e-5	4.63	(4.21)	.002	1.34	(7.59)
`bent-pipe`	10^{-1}	1.54e-01	.060	2.21	(5.43)	.229	1.34	(10.47)
	10^{-3}	1.54e+01	.055	3.68	(8.76)	.034	1.35	(10.08)
	10^{-5}	1.54e+03	.030	3.57	(7.03)	.029	1.47	(14.33)
`recirc`	10^{-1}	1.54e-01	.056	2.21	(5.30)	.259	1.34	(11.44)
	10^{-3}	1.54e+01	.160	3.76	(14.17)	.412	1.34	(38.30)
	10^{-5}	1.54e+03	.173	3.72	(14.65)	.233	1.45	(43.60)

TABLE 4.4

A comparison with classical AMG versus solvers built with αSA LRNK for various instances of Examples 2, 3, and 4 on a 64×64 grid. The AMG methods use V(1,1) cycles with Gauss-Seidel C/F Relaxation. The αSA methods use the approach presented in this paper with different amounts of relaxation for each respective example: V(2,0) cycles for $\gamma = .154$, V(5,0) cycles for $\gamma = 15.4$, and V(7,0) cycles for $\gamma = 1540$ (with the exception of the `recirc` *problem, where V(9,0) cycles were used for $\gamma = 1540$).*

5. Conclusion. This paper presents a nonsymmetric smoothed aggregation approach with several new features. The method is based on a Petrov-Galerkin coarsening that uses approximations to the minimal left and right singular vectors to form restriction and interpolation, respectively. Coarsening uses an aggregation technique that involves a new strength of connection measure. We present a new approach to intergrid transfer operator smoothing that only smoothes in strongly coupled directions within the graph of the problem matrix. Our V-cycles use a nonstationary relaxation, USYMQR [22]. We also present a preliminary two-level convergence result that implies that more relaxation should be used in our framework for nonsymmetric systems than in the SPD setting. The numerical results show that this approach leads to convergent, stand-alone multigrid cycles for many instances of two-dimensional convection-diffusion problems. Moreover, the method is algorithmically scalable for problems that do not have recirculating flow.

Additionally, we present a method to form SA solvers adaptively for nonsymmetric problems, and numerical results show that the adaptive method tends to improve the performance of the multigrid hierarchy. In its current form, nonsymmetric αSA is only implemented to develop primary left and right near-kernel (LRNK) components. However, the long term goal of our efforts is to have an adaptive method that is suitable for nonsymmetric problems that arise from discretizing systems of PDEs, which would involve augmenting the sets of LRNK vectors with secondary kernel.

The numerical results suggest that the multigrid hierarchies formed by classical AMG is more suitable for nonsymmetric problems obtained from discretizing scalar convection-diffusion equations. The intention of our efforts is not to solve all problems more efficiently than classical AMG, but rather to form an adaptive SA method that may be further developed to handle nonsymmetric problems from systems of PDEs, a setting where the adaptive SA methods have excelled for symmetric problems.

REFERENCES

[1] R. Bank and T. Dupont. A comparison of two multilevel iterative methods for nonsymmetric and indefinite elliptic finite element equations. *SIAM J. Numer. Anal.*, 18(4), 1981.

[2] J. Bramble, D. Y. Kwak, and J. Pasciak. Uniform convergence of multigrid v-cycle iterations for indefinite and nonsymmetric problems. In N. D. Melson, T. A. Manteuffel, and S. F. McCormick, editors, *Sixth Copper Mountain Conference on Multigrid Methods*, pages 43–59, 1993.

[3] J. Bramble and J. Pasciak. New convergence estimates for multigrid algorithms. *Mathematics of Computation*, 49(180):311–329, 1987.

[4] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[5] A. Brandt. Algebraic multigrid theory: The symmetric case. *Appl. Math. Comput.*, 9:23–26, 1986.

[6] A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. *DJ Evans (Ed.), Sparsity and its Applications*, 1984.

[7] J. Brannick, M. Brezina, D. Keyes, O. Livne, I. Livshits, S. MacLachlan, T. Manteuffel, S. McCormick, J. Ruge, and L. Zikatanov. Adaptive smoothed aggregation in lattice qcd. In *Domain Decomposition Methods in Science and Engineering XVI*, volume 55 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, 2007.

[8] M. Brezina. Handling of anisotropies in smoothed aggregation code in parSAMIS. presented at the 13th Copper Mountain Conference on Multigrid Methods, March 2007.

[9] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge. Adaptive Smoothed Aggregation (αSA). *SIAM J. on Sci. Comp. (SISC)*, 25:1896–1920, 2004.

[10] M. Brezina, T. Manteuffel, S. McCormick, J. Ruge, G. Sanders, and P. Vassilevski. A generalized eigensolver based on smoothed aggregation (ges-sa) for initializing smoothed aggregation multigrid (sa). *Numerical Linear Algebra with Applications*, 15(2-3):249–269, 2008.

[11] W. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial, 2^{nd} Edition*. SIAM books, 2000.

[12] M. Gee, J. Hu, and R. Tuminaro. A new smoothed aggregation multigrid for anisotropic problems. *to appear in Numerical Linear Algebra*, X(X), X.

[13] H. Guillard and P. Vaněk. An aggregation multigrid solver for convection-diffusion problems on unstructured meshes. Technical Report UCD-CCM-130, Center for Computational Mathematics, University of Colorado, 1998.

[14] U. Hetmaniuk. A Rayleigh quotient minimization algorithm based on algebraic multigrid. *Numerical Linear Algebra with Applications*, 14:563–580, 2007.

[15] J. Mandel. Multigrid convergence for nonsymetric, indefinite variational problems and one smoothing step. *Appl. Math. Comput.*, 19:201–216, 1986.

[16] J. Mandel, S. McCormick, and J. Ruge. An algebraic theory for multigrid methods for variational problems. *SIAM J. Numer. Anal. (SINUM)*, 25(1), 1988.

[17] S. F. McCormick and J. Ruge. Multigrid Methods for Variational Problems. *SIAM J. Numer. Anal.*, 19:925–929, 1982.

[18] J. Ruge. *Multigrid methods for variational and differential eigenvalue problems and unigrid for multigrid simulation*. PhD thesis, Colorado State University, Fort Collins, Colorado, 1981.

[19] J. Ruge and K. Stüben. Algebraic Multigrid (AMG). *Multigrid Methods (McComrick, S.F., ed.)*, 5, 1986.

[20] M. Sala and R. S. Tuminaro. A new petrov-galerkin smoothed aggregation preconditioner for nonsymmetric linear systems. *to appear in SIAM J. on Sci. Comp. (SISC)*, X(X), X.

[21] G. Sanders. *Extensions to Adaptive Smoothed Aggregation Multigrid (αSA): Eigensolver Initialization and Nonsymmetric Problems*. PhD thesis, University of Colorado, Boulder, 2008.

[22] M. A. Saunders, H. D. Simon, and E. L. Yip. Two conjugate-gradient-type methods for unsymmetric linear equations. *SIAM J. Numer. Anal. (SINUM)*, 25(4), 1988.

[23] H. De Sterck, T. A. Manteuffel, S. F. McCormick, J. Pearson, J. Ruge, and G. Sanders. Smoothed aggregation multigrid for markov chains. *to appear in SIAM J. on Sci. Comp. (SISC)*, X, X.

[24] U. Trottenberg, C. W. Osterlee, and A. Schuller (Appendix by K. Stuben). *Multigrid (Appendix A: An Introduction to Algebraic Multigrid)*. Academic Press, 2000.

[25] P. Vaněk. Acceleration of convergence of a two level algorithm by smooth transfer operators. *Appl. Math.*, 37:265–274, 1992.

[26] P. Vaněk, M. Brezina, and J. Mandel. Convergence of algebraic multigrid based on smoothed aggregation. *Numerische Mathematik*, 88:559–579, 2001.

[27] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid on unstructured meshes. Technical Report X, Center for Computational Mathematics, Mathematics Department, 1994.

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

[29] J. Wang. Convergence analysis of multigrid algorithms for nonselfadjoint and indefinite elliptic problems. *SIAM J. Numer. Anal. (SINUM)*, 30(1):275–285, 1993.

[30] I. Yavneh. Coarse-grid correction for nonelliptic and singular perturbation problems. *SIAM J. on Sci. Comp. (SISC)*, 19(5):1682–1699, 1998.

[31] I. Yavneh, C. H. Venner, and A. Brandt. Fast multigrid solution of the advection problem with closed characteristics. *SIAM J. on Sci. Comp. (SISC)*, 1998.