SIAM J. Sci. Comput.
Vol. 33, No. 2, pp. 966–991

A GENERAL INTERPOLATION STRATEGY FOR ALGEBRAIC MULTIGRID USING ENERGY MINIMIZATION[*]

LUKE N. OLSON[†], JACOB B. SCHRODER[‡], AND RAYMOND S. TUMINARO[§]

Abstract. Algebraic multigrid methods solve sparse linear systems $Ax = b$ by automatic construction of a multilevel hierarchy. This hierarchy is defined by grid transfer operators that must accurately capture algebraically smooth error relative to the relaxation method. We propose a methodology to improve grid transfers through energy minimization. The proposed strategy is applicable to Hermitian, non-Hermitian, definite, and indefinite problems. Each column of the grid transfer operator P is minimized in an energy-based norm while enforcing two types of constraints: a defined sparsity pattern and preservation of specified modes in the range of P. A Krylov-based strategy is used to minimize energy, which is equivalent to solving $AP_j = \mathbf{0}$ for each column j of P, with the constraints ensuring a nontrivial solution. For the Hermitian positive definite case, a conjugate gradient (CG-)based method is utilized to construct grid transfers, while methods based on generalized minimum residual (GMRES) and CG on the normal equations (CGNR) are explored for the general case. The approach is flexible, allowing for arbitrary coarsenings, unrestricted sparsity patterns, straightforward long-distance interpolation, and general use of constraints, either user-defined or auto-generated. We conclude with numerical evidence in support of the proposed framework.

Key words. algebraic multigrid, interpolation, nonsymmetric, non-Hermitian, smoothed aggregation

AMS subject classifications. 65F08, 65N55

DOI. 10.1137/100803031

1. Introduction.

Algebraic multigrid (AMG) methods, e.g., classic AMG [5, 21] and smoothed aggregation (SA) [28, 26], are efficient solution algorithms for large, sparse linear systems that often arise in the numerical approximation to partial differential equations (PDEs). These methods automatically construct a multilevel hierarchy by defining grid transfer operators in an effort to effectively combine relaxation, such as weighted-Jacobi, with coarse-grid correction. A key requirement for rapid convergence is that interpolation (i.e., grid transfers) must be accurate for modes invariant to relaxation.

Classical AMG methods were developed for symmetric positive-definite (SPD) systems that arise from elliptic PDEs, and the underlying theory and associated solver components rely on this fact. Extensions to nonsymmetric problems have largely focused on convection-dominated flow which is problematic for standard AMG. Some of

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section July 21, 2010; accepted for publication (in revised form) January 25, 2011; published electronically April 14, 2011. This work was partially supported by the NSF under award DMS-0612448. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/sisc/33-2/80303.html

[†] Siebel Center for Computer Science, University of Illinois at Urbana-Champaign, 201 N. Goodwin Ave., Urbana, IL 61801 (lukeo@illinois.edu).

[‡] Department of Applied Mathematics, University of Colorado at Boulder, UCB 526, Boulder, CO 80309 (jacob.schroder@colorado.edu).

[§] Sandia National Laboratories, P.O. Box 969, MS 9159, Livermore, CA 94551 (rstumin@sandia.gov). Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy under contract DE-AC04-94-AL85000. The work of this author was partially supported by the DOE Office of Science ASCR Program and by the ASC Program at Sandia National Laboratories.

these extensions in the SA context include [9, 25, 22, 27, 12], which have considered using plain or nonsmoothed aggregation for either restriction or for both prolongation and restriction, or have considered interpolating convective and diffusive parts of the discrete operator separately. Restriction refers to interpolation from fine to coarse grids, and prolongation refers to interpolation from coarse to fine. Despite progress, significant issues remain, and AMG for non-Hermitian systems remains an active research area. Energy minimization AMG methods for SPD systems have been proposed [31, 6, 16, 4, 3, 30, 32, 13, 29]. These methods are quite powerful and have a potential to broaden the class of problems for which AMG is successful, because they easily incorporate recent ideas for improving robustness such as compatible relaxation and adaptive multigrid [7, 8, 6]. Energy minimization algorithms, however, have not been adapted to the non-Hermitian case, which is the focus of this paper.

In generalizing energy minimization to non-Hermitian systems, three questions arise. What should be minimized and in what norm? How does restriction differ from prolongation in terms of the first two questions? To answer these, the notion of a left and right near null-space is introduced. In terms of the singular value decomposition (SVD), the left near null-space is represented by left singular vectors corresponding to small singular values, and the right near null-space is likewise represented with respect to right singular vectors. For Hermitian matrices, singular vectors correspond to eigenvectors, and for Hermitian positive-definite (HPD) systems it is generally well understood that accurate interpolation is needed for modes associated with the near null-space. Stated another way, interpolation accuracy for a given mode should be inversely proportional to the Raleigh quotient associated with that mode [17]. In the Hermitian case, the adjoint of prolongation is typically used for restriction as the left and right near null-spaces coincide.

The left and right near null-spaces and their significance within AMG for non-Hermitian systems is discussed in [8, 23]. Consider solving $Ax = b$ using a Richardson iteration

$$x^{(k+1)} = x^{(k)} + \omega(b - Ax^{(k)}),$$

where ω is a damping parameter. The error propagation is given by

$$e^{(k+1)} = (I - \omega A)e^{(k)}, \tag{1.1}$$

with $e^{(k)} = x^{(k)} - A^{-1}b$. If $e^{(k)}$ corresponds to the jth right singular vector with singular value σ_j, then $e^{(k+1)} = e^{(k)} - \omega\sigma_j u_j$, where u_j is the jth left singular vector. When σ_j is small, we have $e^{(k+1)} \approx e^{(k)}$. Moreover, premultiplying (1.1) by u_j^* gives

$$\beta_j^{(k+1)} = \beta_j^{(k)} - \omega\sigma_j v_j^* e^{(k)},$$

where $\beta_j^{(k)} = u_j^* e^{(k)}$ and v_k is the associated right singular vector. If σ_j is small, then the implication is that error components in the direction of u_j are essentially undamped. It follows that the coarse level correction must address both the left and right near null-spaces as relaxation is not effective. Correspondingly, the energy minimization proposed in this paper targets the accurate grid transfer of singular vectors associated with small singular values [8, 23]. Specifically, the range of prolongation must accurately represent right singular vectors associated with small singular values, while the range of the adjoint of restriction must accurately represent left singular vectors associated with small singular values. Moreover, we also propose constructing the restriction operators through a suitable algorithm for constructing prolongation

[2, 10, 23, 15]. In our context, the algorithm to construct prolongation is used with the adjoint of the discrete operator yielding a surrogate prolongation operator. Restriction (and only restriction) is then defined as the adjoint of this surrogate. Therefore, only prolongation algorithms are presented, but the numerical results use the proposed algorithms to also construct the restriction operators.

From a non-Hermitian AMG perspective, this paper is most closely related to [8, 23]. In [23] a method is proposed based on smoothing restriction and prolongation basis functions with the heuristic goal of reducing their energy in an AA^*-norm and an A^*A-norm, respectively. However, the method is limited to one smoothing step and does not fully consider the left and right near null-spaces. In [8], an approximation property is shown for non-Hermitian systems which guarantees two-level mesh-independent convergence. Yet, construction of R and P is based on an expensive process that uses Richardson's method on AA^* and A^*A. Additionally, the method does not facilitate multiple near null-space modes which is necessary for some problems, e.g., elasticity.

From an energy minimization perspective, this paper extends ideas for Hermitian systems to non-Hermitian matrices [31, 6, 16]. These earlier approaches reduce the energy of prolongator basis functions in an energy-related norm via an iterative procedure. A key component is that minimization is done while satisfying constraints. These constraints force the range of prolongation to include user-provided near null-space modes in addition to enforcing a preset sparsity pattern. The process results in standard SA prolongation smoothing when only one iteration is used [16]; this is also related to an extension of [16] to a conjugate gradient (CG-)based approach by Vaněk (unpublished). Additionally, [18] developed a conjugate gradient normal residual (CGNR-)based prolongation smoothing scheme, which is further developed here.

The proposed general energy minimization approach to AMG has several advantages in terms of flexibility. These include allowing for arbitrary coarsenings and sparsity patterns, accurate interpolation of multiple near null-space modes, and the use of general constraints. In addition, the method allows for straightforward long-distance interpolation and improved control of complexity—i.e., cost per iteration. The framework also enables more recent approaches to be easily incorporated, including compatible relaxation and adaptive multigrid. Finally, the framework is attractive since it is applicable to both SA and classic AMG, although we use SA terminology and carry out most experiments with SA.

In section 2, we introduce the general energy minimization framework and the several specific Krylov-based strategies explored here. In section 3, we discuss numerical considerations, such as initial feasible prolongators and the conditioning of prolongator columns, for both SA-based and F/C-based AMG. In section 4, we discuss aspects of implementation and computational cost. In section 5, we present supporting numerical evidence for the proposed energy minimization framework. In section 6, concluding remarks are made.

2. Energy minimization.

In this section, we present an energy minimization framework that generalizes ideas for Hermitian systems and additionally leads to algorithms applicable to the non-Hermitian case. Energy minimization algorithms such as SA [28, 26], trace minimization AMG [6], and those in [31, 16] all minimize energy for the Hermitian case of prolongator basis functions subject to constraints. Stated in norm χ we have

$$P = \underset{P}{\operatorname{argmin}} \sum_j \|P_j\|_\chi^2 \quad \text{and} \quad PB^C = B, \text{ with } P \in \mathcal{N}, \tag{2.1}$$

where P_j is the jth column of P, B (B^C) is a column matrix of fine (coarse) level near null-space modes, and $P \in \mathcal{N}$ specifies that P's sparsity pattern conforms to a specified pattern, $\mathcal{N}$. The dimensions of P, B^C, and B are $n \times q$, $q \times m$, and $n \times m$, respectively. So, m defines the number of near null-space modes that P must exactly interpolate.

Our approach is to utilize the minimization property of Krylov methods in seeking a solution to (2.1). Consider finding $\Delta\check{P}$ in the linear system

$$(I_q \otimes A)\Delta\check{P} = (I_q \otimes A)\check{P}^{(0)}, \tag{2.2}$$

where A and $\check{P}^{(0)}$ are given. Here, A is $n \times n$, I_q is the $q \times q$ identity matrix, and the operator $\check{}$ denotes columnwise conversion of a matrix to vector form. That is,

$$H = [H_1 \dots H_s], \text{ where } H_j \text{ is the } j\text{th column}, \qquad \text{and} \qquad \check{H} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_s \end{bmatrix}. \tag{2.3}$$

Equation (2.2) has the trivial solution $\Delta\check{P} = \check{P}^{(0)}$, but we instead consider an approximation obtained by a Krylov method such as flexible GMRES (fGMRES) with a limited or restricted search space. At the ith Krylov iteration, the current approximation $\Delta\check{P}^{(i)}$ corresponds to an optimal solution residing in the given search space $\mathcal{K}_i$, i.e.,

$$\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i = \operatorname{span}\{\check{V}^{(1)}, \dots, \check{V}^{(i)}\}, \tag{2.4}$$

where $\check{V}^{(i)}$ is the ith Krylov vector. If $\check{P}^{(0)} \in \check{\mathcal{K}}_i$, then fGMRES converges to the trivial solution. Suppose, however, that the search space is restricted to satisfy h constraints given by the $h \times nq$ matrix X, and that $\check{P}^{(0)} \notin \check{\mathcal{K}}_i$. That is,

$$\forall k \leq i, \quad X\check{V}^{(k)} = \mathbf{0} \text{ and that } X\check{P}^{(0)} = \begin{bmatrix} \mathbf{0} \\ \check{B} \end{bmatrix}. \tag{2.5}$$

The constraints imply that $X\Delta\check{P}^{(i)} = \mathbf{0}$ and so (2.2) cannot be solved. That is, the residual is never reduced below a certain threshold. The matrix X is discussed in the next section and is chosen to enforce both the sparsity pattern and the $PB^C = B$ constraint. The underlying idea is that (2.5) defines an initial constraint-satisfying feasible solution $P^{(0)}$ and ensures that a prolongator given by $P = P^{(0)} - \Delta P$ also satisfies the constraints.

Iteratively, a method such as fGMRES minimizes the Euclidean norm of the ith residual over all possible solutions in $\check{\mathcal{K}}_i$. The residual is

$$\check{R}^{(i)} = (I_q \otimes A)\check{P}^{(0)} - (I_q \otimes A)\Delta\check{P}^{(i)} \tag{2.6a}$$

$$= (I_q \otimes A)\check{P}^{(i)}, \tag{2.6b}$$

where $\check{P}^{(i)} = \check{P}^{(0)} - \Delta\check{P}^{(i)}$. Thus, fGMRES effectively computes, due to the block structure of $(I_q \otimes A^*A)$,

$$\Delta\check{P}^{(i)} = \underset{\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i}{\operatorname{argmin}} \sum_{j=1}^{q} \left\| P_j^{(i)} \right\|_{A*A}^2 \tag{2.7}$$

$$= \underset{\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i}{\operatorname{argmin}} \left\| \check{P}^{(i)} \right\|_{I_q \otimes A^*A}^2, \quad \text{with} \quad X\check{P}^{(i)} = \begin{bmatrix} \mathbf{0} \\ \check{B} \end{bmatrix}. \tag{2.8}$$

Thus, fGMRES realizes (2.1) with $\chi = A^*A$ and minimization restricted to $P - P^{(0)} \in \mathcal{K}_i$. An "ideal" choice for the search space is

$$\mathcal{K}_i = \check{V}^{(k)} = Q(I_q \otimes A)^k \check{P}^{(0)}, \tag{2.9}$$

where

$$Q = I - X^*(XX^*)^{-1}X. \tag{2.10}$$

For now, we assume that Q is uniquely defined, i.e., X is full rank. Section 2.1.1 describes how rank deficiencies in X arise, and section 4 describes a robust implementation for our constraint choices that guarantees a unique definition of Q.

We consider this an ideal choice because this space is only slightly modified from a standard Krylov space. Specifically, there is only one application of Q. Unfortunately, this fGMRES approach is prohibitively expensive. Not only is fGMRES more expensive than other Krylov methods (e.g., GMRES), but calculation of the intermediate vectors $(I_q \otimes A)^k \check{P}^{(0)}$ includes many nonzeros. It is only after multiplication by Q that the sparsity pattern is restricted to $\mathcal{N}$. The key point, however, is that other computationally attractive Krylov methods are also applicable to (2.2), resulting in (2.1) for different norms and search spaces.

2.1. Constraints. The matrix X is chosen to enforce both $P^{(i)} \in \mathcal{N}$ and $P^{(i)}B^C = B$. To illustrate this, we consider the following example.

Example 2.1. Consider a 4×2 prolongation operator and let $P = P^{(i)}$. Specifically, let

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} \\ p_{2,1} & p_{2,2} \\ p_{3,1} & p_{3,2} \\ p_{4,1} & p_{4,2} \end{bmatrix}, \quad B = \begin{bmatrix} b_{1,1} \\ b_{2,1} \\ b_{3,1} \\ b_{4,1} \end{bmatrix}, \quad \text{and} \quad B^C = \begin{bmatrix} b^c_{1,1} \\ b^c_{2,1} \end{bmatrix}. \tag{2.11}$$

Additionally, define P's sparsity pattern by

$$\mathcal{N} = \begin{bmatrix} * & * \\ * & 0 \\ * & * \\ 0 & * \end{bmatrix} \tag{2.12}$$

and impose the constraint $PB^C = B$. In this case,

$$\check{P} = \begin{bmatrix} p_{1,1} & p_{2,1} & p_{3,1} & p_{4,1} & p_{1,2} & p_{2,2} & p_{3,2} & p_{4,2} \end{bmatrix}^T, \tag{2.13}$$

and (2.5) becomes

$$X\check{P} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ b^c_{1,1} & 0 & 0 & 0 & b^c_{2,1} & 0 & 0 & 0 \\ 0 & b^c_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{1,1} & 0 & 0 & 0 & b^c_{2,1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & b^c_{2,1} \end{bmatrix}, \quad \check{P} = \begin{bmatrix} 0 \\ 0 \\ b_{1,1} \\ b_{2,1} \\ b_{3,1} \\ b_{4,1} \end{bmatrix}. \tag{2.14}$$

The first two constraint rows enforce the sparsity pattern, while the last four enforce $PB^C = B$. If B^C is instead 2×2 and B is 4×2, the last four constraints are replicated with $b^c_{i,2}$ replacing $b^c_{i,1}$ and $b_{i,2}$ replacing $b_{i,1}$.

2.1.1. General case. Let $\mathcal{N}$ contain a total of r nonzeros. Define $I : \mathbb{C}^r \rightarrow \mathbb{C}^{nq}$ as the injection operator given by

$$I = \begin{bmatrix} I_{(1)} & 0 & \dots & \\ 0 & I_{(2)} & 0 & \dots \\ & & \ddots & \\ & \dots & 0 & I_{(q)} \end{bmatrix}, \tag{2.15}$$

where $I_{(j)}$ is obtained by taking an $n \times n$ identity matrix and removing columns associated with entries excluded from the jth column of $\mathcal{N}$. Thus, $I^T \check{P}$ is an $r \times 1$ vector corresponding to P's nonzeros. In Example 2.1,

$$I^T \check{P} = \begin{bmatrix} p_{1,1} & p_{2,1} & p_{3,1} & p_{1,2} & p_{3,2} & p_{4,2} \end{bmatrix}^T .$$

The injection operator is used to reduce the dimension of X by excluding zero entries. In Example 2.1, the reduced constraint matrix X is obtained by removing the first two rows and the 4th and 6th columns from (2.14). In general, $X = (B^C \otimes I_n)^T I$, which yields the reduced form of the constraint equations

$$X I^T \Delta\check{P} = \begin{bmatrix} b^c_{1,1} I_{(1)} & b^c_{2,1} I_{(2)} & \dots & b^c_{q,1} I_{(q)} \\ b^c_{1,2} I_{(1)} & b^c_{2,2} I_{(2)} & \dots & b^c_{q,2} I_{(q)} \\ \vdots & \vdots & \vdots & \vdots \\ b^c_{1,m} I_{(1)} & b^c_{2,m} I_{(2)} & \dots & b^c_{q,m} I_{(q)} \end{bmatrix} I^T \Delta\check{P}. \tag{2.16}$$

Notice that the reduced constraint matrix X in (2.16) can be permuted to block diagonal form where each block corresponds to the allowed nonzeros for a given row of P. Now if each $I_{(j)}$ is an identity, i.e., P is fully dense, each diagonal block is the transpose of B^C:

$$\begin{bmatrix} b^c_{1,1} & b^c_{2,1} & \dots & b^c_{q,1} \\ b^c_{1,2} & b^c_{2,2} & \dots & b^c_{q,2} \\ \vdots & \vdots & & \vdots \\ b^c_{1,m} & b^c_{2,m} & \dots & b^c_{q,m} \end{bmatrix}. \tag{2.17}$$

In practice, each $I_{(j)}$ is not a full identity and again reflects removal of the columns associated with entries excluded from $\mathcal{N}$. In Example 2.1, X permuted to this block diagonal form is

$$\begin{bmatrix} b^c_{1,1} & b^c_{2,1} & & & & \\ & & b^c_{1,1} & & & \\ & & & b^c_{1,1} & b^c_{2,1} & \\ & & & & & b^c_{2,1} \end{bmatrix}. \tag{2.18}$$

In general, if a row rank exceeds a column rank within any of the individual diagonal blocks, then for the general case, the constraints can only be satisfied in a least-squares sense. Furthermore, linear dependence between rows of the individual diagonal blocks requires the introduction of a pseudoinverse. It should be noted that the diagonal blocks give the permuted X a local nature that allows for constraint satisfaction to be implemented efficiently in parallel.

Last, we show how during practical computations only reduced forms of the above operators $(I_q \otimes A)$, Q, and X are necessary. Notice that Q and II^T contain a zero row/column corresponding to each entry in P excluded from the sparsity pattern (this would be rows/columns 4 and 6 in Example 2.1) and that $\check{P}^{(0)}$ is also zero for entries excluded from the sparsity pattern (rows 4 and 6 in Example 2.1). It follows that $Q = II^TQII^T$ and $\check{P}^{(0)} = II^T\check{P}^{(0)}$. Additionally, define the permutation matrix $[\bar{I},\, I\,]$, where $\bar{I}$ are the columns of the identity excluded when constructing I and

$$X[\bar{I},\, I\,] = \begin{bmatrix} I & 0 \\ 0 & X \end{bmatrix}.$$

Thus, the "ideal" fGMRES Krylov space (2.9) is

$$\check{V}^{(k)} = II^TQII^T(I_q \otimes A)^k II^T\check{P}^{(0)} \tag{2.19a}$$

$$= IQ(I_q \otimes A)\Delta\check{P}, \tag{2.19b}$$

where

$$\Delta\check{P} = I^T{\scriptstyle\Delta}\check{P}, \tag{2.19c}$$

$$Q = I^TQI \tag{2.19d}$$

$$= I - I^TX^*\left(X[\bar{I},\, I\,]\,[\bar{I},\, I\,]^TX^*\right)^{-1}XI \tag{2.19e}$$

$$= I - X^*\left(XX^*\right)^{-1}X, \tag{2.19f}$$

and

$$I_q \otimes A = I^T(I_q \otimes A)^k I \tag{2.19g}$$

$$= \begin{bmatrix} I_{(1)}^T A^k I_{(1)} & & 0 \\ & \ddots & \\ 0 & & I_{(q)}^T A^k I_{(q)} \end{bmatrix}. \tag{2.19h}$$

Overall, this implies that one need not work with the full operators in practical computations but instead use only the reduced forms of $(I_q \otimes A)$, Q, and X.

In summary, the injection operator I enforces the sparsity pattern, while the reduced constraint matrix X enforces $PB^C = B$. Moreover, the minimization by fGMRES in the norm induced by $(I_q \otimes A^*A)$ is equivalent to minimization in the norm induced by $(I_q \otimes A^*A)$. For simplicity, we drop the notation for the reduced operators throughout the rest of the paper, noting that for practical computations one works with the reduced operators.

2.2. *QA*-GMRES variant. Consider premultiplication of (2.2) by Q followed by application of GMRES. That is,

$$Q_A{\scriptstyle\Delta}\check{P} = Q_A\check{P}^{(0)} \tag{2.20a}$$

with

$$Q_A = Q(I_q \otimes A), \tag{2.20b}$$

$${\scriptstyle\Delta}\check{P}^{(0)} = \mathbf{0}, \tag{2.20c}$$

$${\scriptstyle\Delta}\check{P}^{(i)} \in \check{\mathcal{K}}_i = \operatorname{span}\{Q_A\check{P}^{(0)}, Q_A^2\check{P}^{(0)} \ldots Q_A^i\check{P}^{(0)}\}. \tag{2.20d}$$

The ith residual is given by

$$\check{R}^{(i)} = Q_A(\check{P}^{(0)} - \Delta\check{P}^{(i)}) = Q_A\check{P}^{(i)}. \tag{2.21}$$

Since GMRES minimizes the Euclidean norm of the residual, this approach achieves

$$\min_{\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i} \left\| \check{P}^{(i)} \right\|^2_{Q_A^* Q_A}, \tag{2.22}$$

where $\| \cdot \|^2_{Q_A^* Q_A}$ is a seminorm in general, but a full norm over $\check{\mathcal{K}}_i$. Unlike $(I_q \otimes A)$, Q is not block diagonal, so the definition of energy cannot be split into the sum of the energies of individual columns. It is important to notice that despite the singularity associated with Q in (2.20a), the solution is still unique, as summarized in the following lemma.

Lemma 2.2. *If A is nonsingular, QA-GMRES converges to a unique solution of* (2.20a) *satisfying* (2.22).

Proof. Let S be an orthonormal basis for the null-space of Q and let $S_\perp$ be an orthonormal basis for its complement. That is,

$$S_\perp^* S_\perp = I, \quad S^* S = I,$$
$$S^* S_\perp = 0, \quad QS = 0,$$

as well as

$$S_\perp S_\perp^* = Q. \tag{2.23}$$

The last equality follows from the fact that both $S_\perp S_\perp^*$ and Q correspond to the unique orthonormal projector onto the space associated with $S_\perp$.

Consider now the linear system

$$S_\perp^*(I_q \otimes A)S_\perp y = S_\perp^*(I_q \otimes A)\check{P}^{(0)}. \tag{2.24}$$

Since $I_q \otimes A$ is nonsingular, it follows that $S_\perp^*(I_q \otimes A)S_\perp$ is nonsingular, so the system has a unique solution given by

$$y = (S_\perp^*(I_q \otimes A)S_\perp)^{-1} S_\perp^*(I_q \otimes A)\check{P}^{(0)},$$

which can be computed by applying GMRES to (2.24) starting with a zero initial guess. That is, GMRES is guaranteed to converge due to the nonsingularity of $S_\perp^*(I_q \otimes A)S_\perp$, and the unique solution lies in the Krylov space

$$\begin{aligned} \operatorname{span}\{&S_\perp^*(I_q \otimes A)\check{P}^{(0)}, (S_\perp^*(I_q \otimes A)S_\perp)\, S_\perp^*(I_q \otimes A)\check{P}^{(0)}, \ldots, \\ &(S_\perp^*(I_q \otimes A)S_\perp)^k\, S_\perp^*(I_q \otimes A)\check{P}^{(0)}\}, \end{aligned}$$

where k is the number of GMRES iterations required for convergence. However, premultiplication of (2.24) by $S_\perp$ and application of (2.23) reveals that $S_\perp^* y$ is a solution to (2.20a). Further, premultiplication of the Krylov space for y by $S_\perp$ and application again of (2.23) reveals that $S_\perp^* y$ lies in the Krylov space given by (2.20d). This concludes the proof as $S_\perp^* y$ is a unique solution that lies in the Krylov space associated with GMRES applied to (2.20a), and minimizes the residual. □

The QA-GMRES algorithm is attractive since the search space is similar to that of a standard Krylov space. The difference with fGMRES is that Q is now directly inserted into the system (2.20a), so that in (2.20d), every application of $(I_q \otimes A)$ is followed by a Q. This controls the complexity by limiting the sparsity pattern of the intermediate products for each Krylov vector in (2.20d). This aspect of QA-GMRES and all subsequent approaches allows for the construction of computationally feasible methods. The minimization norm also differs slightly due to the presence of Q. This term ensures that GMRES attains a zero residual (assuming A is nonsingular) as opposed to the fGMRES case. It does this by effectively ignoring energy associated with $(I_q \otimes A)\check{P}$ that violates the constraints.

2.3. *QAQ*-CG variant. Assume that A is HPD and consider the application of CG to

$$Q_A Q \Delta\check{P} = Q_A \check{P}^{(0)} \tag{2.25a}$$

with

$$\Delta\check{P}^{(0)} = \mathbf{0}, \tag{2.25b}$$

$$\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i = \operatorname{span}\{Q_A\check{P}^{(0)}, Q_A^2\check{P}^{(0)} \ldots Q_A^i\check{P}^{(0)}\}. \tag{2.25c}$$

The Q immediately preceding $\Delta\check{P}$ in (2.25a) is inserted for symmetry but is actually not needed computationally, since the nature of the Krylov space implies that $\Delta\check{P}^{(i)} = Q\Delta\check{P}^{(i)}$ for each iteration i.

The ith residual is given by

$$\check{R}^{(i)} = Q_A(\check{P}^{(0)} - Q\Delta\check{P}^{(i)}) = Q_A\check{P}^{(i)}. \tag{2.26}$$

Lemma 2.3. *Let A be HPD. Then CG applied to* (2.25a) *with a zero initial guess computes*

$$\Delta\check{P}^{(i)} = \underset{\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i}{\operatorname{argmin}} \left\| \check{P}^{(i)} \right\|^2_{I_q \otimes A}, \tag{2.27}$$

where $\mathcal{K}_i$ is the space in (2.25c).

Proof. Despite Q_AQ being Hermitian positive semidefinite, Q_AQ still induces an energy-norm over span(Q_AQ), and in particular over $\mathcal{K}_i$. Additionally, we utilize the fact that CG minimizes the error in this energy-norm for consistent singular linear systems with an initial residual in the span of the operator.

Let $\check{P}^{\star}$ be a solution to (2.25a). Then CG computes

$$\Delta\check{P}^{(i)} = \underset{\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i}{\operatorname{argmin}} \left\| \check{P}^{\star} - \Delta\check{P}^{(i)} \right\|^2_G, \tag{2.28a}$$

where $G = Q_AQ$. Together with $\check{P}^{\star}$ satisfying (2.25a) and using $Q\Delta\check{P}^{(i)} = \Delta\check{P}^{(i)}$, the square of the norm leads to

$$\underset{\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i}{\operatorname{argmin}} \left\| \Delta\check{P}^{(i)} \right\|^2_G + \left\| \check{P}^{\star} \right\|^2_G - \langle \Delta\check{P}^{(i)},\ (I_q \otimes A)\check{P}^{(0)} \rangle - \langle \check{P}^{(0)},\ (I_q \otimes A)\Delta\check{P}^{(i)} \rangle. \tag{2.28b}$$

Since $\left\| \check{P}^{\star} \right\|^2_G$ does not affect the minimum, we replace it with $\left\| \check{P}^{(0)} \right\|^2_G$, which concludes the proof since

$$\underset{\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i}{\operatorname{argmin}} \left\| \check{P}^{(0)} - \Delta\check{P}^{(i)} \right\|^2_{I_q \otimes A} = \Delta\check{P}^{(i)}. \qquad \square \tag{2.28c}$$

Corollary 2.4. *Given the block structure of* $(I_q \otimes A)$*, the QAQ-CG approach minimizes* $P^{(i)}$ *columnwise—i.e.,*

$$\Delta\check{P}^{(i)} = \underset{\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i}{\operatorname{argmin}} \sum_{j=1}^{q} \left\| P_j^{(i)} \right\|_A^2 . \tag{2.29}$$

Additionally, we conclude that the solution found by CG is unique.

Lemma 2.5. *Assume A is HPD; then*

$$\check{P}^{\star} = U \left(U^{*}(I_q \otimes A)\, U \right)^{-1} U^{*}\alpha + U_z\gamma \tag{2.30}$$

are solutions to (2.25a)*.* U_z *is an* $nq \times \ell$ *basis for the null-space of* Q*, and* U *is an* $nq \times (nq - \ell)$ *basis for the null-space of* $I - Q$*. Further, the solution computed by CG for* (2.25a) *is unique and corresponds to* $\gamma = \mathbf{0}$ *and* $\alpha = (I_q \otimes A)\check{P}^{(0)}$*.*

Proof. The solution is verified by premultiplication of (2.25a) with the nonsingular matrix $[U, U_z]^{*}$. As the Krylov space $\mathcal{K}$ used by CG has no component in the direction of U_z, $\gamma = \mathbf{0}$ follows. □

This is an attractive algorithm when A is Hermitian because of the 3-term recurrence. Also, minimization occurs in a norm for each column, which is similar to fGMRES and unlike QA-GMRES.

2.4. $QA^{*}AQ$-CGNR variant.

The $QA^{*}AQ$-CGNR variant was previously presented in [18], but we now provide a more thorough investigation. Consider using CG to solve

$$Q_{A^{*}A} Q\, \Delta\check{P} = Q_{A^{*}A}\check{P}^{(0)} \tag{2.31a}$$

with

$$Q_{A^{*}A} = Q(I_q \otimes A^{*}A), \tag{2.31b}$$

$$\Delta\check{P}^{(0)} = \mathbf{0}, \tag{2.31c}$$

$$\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i = \operatorname{span}\{Q_{A^{*}A}P^{(0)}, Q_{A^{*}A}^{2}P^{(0)} \ldots Q_{A^{*}A}^{i}P^{(0)}\}. \tag{2.31d}$$

Similar to QAQ-CG, the Q immediately preceding $\Delta\check{P}$ in (2.31a) is inserted for symmetry but is never implemented. From (2.31a), the ith residual is given by

$$R^{(i)} = Q_{A^{*}A}(P^{(0)} - Q\Delta P^{(i)}) = Q_{A^{*}A}P^{(i)}, \tag{2.32}$$

and we arrive at the following CGNR result.

Lemma 2.6. *Let A be nonsingular. Then CGNR applied to* (2.31a) *with a zero initial guess computes*

$$\Delta\check{P}^{(i)} = \underset{\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i}{\operatorname{argmin}} \left\| \check{P}^{(i)} \right\|_{I_q \otimes A^{*}A}^2 . \tag{2.33}$$

Proof. The result follows from Lemma 2.3. □

Corollary 2.7. *Given the block structure of* $(I_q \otimes A^{*}A)$*, the* $QA^{*}AQ$*-CGNR approach minimizes* $P^{(i)}$ *columnwise—i.e.,*

$$\Delta\check{P}^{(i)} = \underset{\Delta\check{P}^{(i)} \in \check{\mathcal{K}}_i}{\operatorname{argmin}} \sum_{j=1}^{q} \left\| P_j^{(i)} \right\|_{A^{*}A}^2 . \tag{2.34}$$

Additionally, we conclude that the solution found by CGNR is unique.

LEMMA 2.8. *Assume A is HPD; then*

$$\check{P}^{\star} = U\left(U^{*}(I_q \otimes A^{*}A)\,U\right)^{-1} U^{*}\alpha + U_z\gamma \tag{2.35}$$

are solutions to (2.31a). *U_z is an $nq \times \ell$ basis for the null-space of Q, and U is an $nq \times (nq-\ell)$ basis for the null-space of $I-Q$. Further, the solution computed by CGNR for* (2.31a) *is unique and corresponds to $\gamma = \mathbf{0}$ and $\alpha = (I_q \otimes A^{*}A)\check{P}^{(0)}$.*

Proof. The result follows from Lemma 2.5. □

A benefit of this approach is that minimization occurs in a norm for each column j, unlike QA-GMRES. Additionally, the Krylov method has short recurrences. However, the $A^{*}A$-like Krylov space raises conditioning concerns as well as potential problems for some differential operators such as convection-diffusion. In particular,

$$Au \approx u_{xx} + u_x \Rightarrow \tag{2.36a}$$

$$A^{*}Au \approx u_{xxxx} + u_{xx}. \tag{2.36b}$$

This implies that the convection term does not play a role in the prolongator choice which intuitively might be suboptimal.

3. Numerical considerations.

Numerical issues associated with energy minimization are now discussed. In the next section, implementation and computational considerations are explored.

3.1. Initial feasible solution.

A matrix $\check{P}^{(0)}$ which satisfies $X\check{P}^{(0)} = \check{B}$ is required by the energy minimization algorithms.[1] The operator X in turn requires the coarse near null-space to be defined. We discuss possible $\check{P}^{(0)}$ and B^{C} choices for F/C-based and aggregation-based AMG.

Classical AMG algorithms proceed by partitioning all fine degrees-of-freedom into two subsets (F-points and C-points) and requiring that interpolation at C-points correspond to injection. That is,

$$P^{(0)} = \begin{bmatrix} P_F \\ I \end{bmatrix}, \tag{3.1}$$

where degrees-of-freedom are ordered so that all F-points precede all C-points. This effectively defines the coarse near null-space as the fine level near null-space injected to the coarse level, and so X is now defined via (2.16). A P_F satisfying the constraints is determined in a number of different ways. One general approach that can be used with *any* initial prolongator guess is

$$\check{P}^{(0)} \leftarrow \check{P}^{(0)} + X^{*}(XX^{*})^{-1}(\check{B} - X\check{P}^{(0)}), \tag{3.2}$$

since premultiplication by X trivially demonstrates that $X\check{P}^{(0)} = \check{B}$. Implementation corresponds to a minor modification of the primary kernel associated with application of Q. In the case of (3.1), it is sufficient to choose an initial P_F of zero followed by an application of (3.2), which preserves the F/C form of (3.1).

Alternatively, aggregation algorithms construct a block diagonal prolongator that also trivially satisfies constraints. Specifically, SA generates a tentative prolongator

[1] This slight modification of (2.5) is due to the implicit use of reduced operators discussed in section 2.1.1 where sparsity pattern constraints are implicitly satisfied by injection operators. For example, $\check{B}$ now replaces the $[\mathbf{0},\, \check{B}^{T}]^{T}$ used previously.

and coarse near null-space

$$P^{(0)} = \begin{bmatrix} Q_{(1)} & 0 & \dots & \\ 0 & Q_{(2)} & 0 & \dots \\ & & \ddots & \\ & \dots & 0 & Q_{(k)} \end{bmatrix}, \quad B^C = \begin{bmatrix} R_{(1)} \\ R_{(2)} \\ \ddots \\ R_{(k)} \end{bmatrix}, \tag{3.3}$$

where $B_{(i)}$ is a submatrix of B obtained by taking only rows assigned to the ith aggregate and $B_{(i)} = Q_{(i)}R_{(i)}$ via a QR factorization. It follows that (3.3) satisfies the initial constraint as well.

3.2. SA-based energy minimization. In this section, we give a brief overview of SA extensions for non-Hermitian problems. These extensions require an initial feasible $P^{(0)}$, a coarse level near null-space representation, and a sparsity pattern in order to apply an energy minimization algorithm.

The recursive solve phase of SA-based AMG follows that of standard multigrid. Presmoothing is done on the fine level in order to reduce high-energy modes in the error. The residual is then transferred to a coarse grid, and the coarse residual problem is solved recursively until the current coarse grid operator has a trivial number of degrees-of-freedom and is solved directly. The fine grid solution is then corrected using the interpolated coarse-grid solution, followed by postsmoothing.

Algorithm 1: `sa_setup`$(A_0,\ B_0^L, B_0^R)$

```
for $lev = 0$ to $lev_{max}$ do
    smooth $\eta$ times on $A_{lev}B_{lev}^R = \mathbf{0}$
    smooth $\eta^*$ times on $A_{lev}^*B_{lev}^L = \mathbf{0}$

    $S_{lev}$ = strength($A_{lev}$)
    $\mathcal{N}_{lev}$ = aggregate($S_{lev}$)

    $P_{lev}^{(0)}, B_{lev+1}^R$ = inject_modes($\mathcal{N}_{lev}, B_{lev}^R$)
    $P_{lev}$ = prolongator_smoother($A_{lev}, P_{lev}^{(0)}$)

    if $A_{lev}$ is non-Hermitian then
        $R_{lev}^{(0)*}, B_{lev+1}^L$ = inject_modes($\mathcal{N}_{lev}, B_{lev}^L$)
        $R_{lev}^*$ = prolongator_smoother($A_{lev}^*, R_{lev}^{(0)*}$)
    else
        $R_{lev} = P_{lev}^*$

    $A_{lev+1} = R_{lev}A_{lev}P_{lev}$
return $A_0, \dots, A_{lev}, P_0, \dots, P_{lev-1}, R_0, \dots, R_{lev-1}$
```

Algorithm 1 outlines our approach to an SA-based construction of P and R for non-Hermitian problems. P_j interpolates from grid $j+1$ to grid j, R_j restricts from grid j to grid $j+1$, and A_j is the operator on grid j. Subscripts will be suppressed when the discussion is invariant to the grid level. To begin, we assume that a collection of right and left near null-space modes, B^R and B^L, are provided by the user, either through knowledge of the underlying problem, or by an adaptive procedure [7]. The left and right near null-spaces are not equivalent, necessitating separate procedures for interpolation and restriction. As motivated in section 1, we want span(R^*) and span(P) to approximate the left and right near null-spaces of A, respectively. We employ the construction procedure based on A for P to define an R^* based on A^*. B^R and B^L are often rough representations of the near null-space, and so smoothing by A and A^*, respectively, improves these so that they more accurately mimic the near null-space, particularly at domain boundaries.

Classical SA first generates a coarse near null-space representation and a simple tentative prolongator to accurately interpolate near null-space modes as sketched in (3.3). This tentative prolongator $P^{(0)}$ is block diagonal where each block corresponds to all degrees-of-freedom within an aggregate. Aggregation corresponds to a nonoverlapping covering of a matrix graph (i.e., assigning each graph vertex to an aggregate) and is often accomplished through a greedy procedure. While the matrix A can be used to define a matrix graph, we instead aggregate based on the graph derived from a strength matrix, S, which relates the ability to interpolate smooth error between two degrees-of-freedom j_1 and j_2 in the matrix graph of A [19]. Classical SA then improves upon the tentative prolongator by performing a smoothing procedure which is usually one iteration of weighted-Jacobi on the equation $AP^{(0)} = \mathbf{0}$. This lowers the energy of each column in P, thus increasing accuracy of low-energy modes. Each Jacobi sweep also extends the prolongator sparsity pattern, and the method becomes prohibitively expensive if the prolongator sparsity pattern is too dense. One smoothing step yields acceptable sparsity patterns when aggregates of diameter three are generated in the aggregation phase (as is normally done). It is important to notice that after the smoothing sweep the prolongator continues to interpolate the near null-space when the near null-space is actually an exact null-space. As this is not normally the case, the smoothed prolongator only approximately interpolates the near null-space.

Our SA-based energy minimization mirrors classical SA with respect to the tentative prolongator $P^{(0)}$ and the coarse representation of the near null-space. Additionally, a tentative $R^{(0)}$ is constructed using B^L in an analogous fashion. The sparsity pattern $\mathcal{N}$ for energy minimization corresponds to $|S|^d|P^{(0)}|$ when aggregate diameters are $2^d + 1$ (where $|\cdot|$ denotes an elementwise absolute value). We normally use aggregates with a diameter of three so that in this case $d = 1$. This also mirrors classical SA, which has the identical sparsity pattern when d Jacobi sweeps are performed. It is important to note that more general sparsity patterns are possible with energy minimization, but these have not yet been explored.

Another important feature is that if B^R and B^L are presmoothed only on the finest level, then the resulting grid transfers satisfy $P_{lev}B^R_{lev+1} = B^R_{lev}$ and $R^*_{lev}B^L_{lev+1} = B^L_{lev}$ for each level in the hierarchy. That is, span($P_0P_1 \dots P_{lev_{max}}$) exactly preserves B^R_0, and span($R^*_0R^*_1 \dots R^*_{lev_{max}}$) exactly preserves B^L_0. This is also an important difference with classic SA, and it is for this reason that presmoothing the left and right near null-space representations is important as they are exactly represented throughout the hierarchy. For instance, this presmoothing addresses the fact that a standard near null-space (such as the constant or rigid body modes) is typically inaccurate when close to domain boundaries, and this affects the quality of the resulting prolongator. For many problems, this presmoothing of the near null-space is unnecessary, but we note that for the case of the Helmholtz problem [18] and some three-dimensional elasticity problems, prerelaxing B is beneficial. Moreover, presmoothing the near null-space modes at each level also provides improvement and approaches a purely adaptive method. It should be noted that the presmoothing presented here is an alternative to an approach outlined in [16] where constraints are ignored near boundaries. An additional advantage of Algorithm 1 is that for non-Hermitian matrices, accuracy is improved iteratively and is not limited to a single Jacobi sweep.

Intuitively, a complementary relationship between grid transfers and relaxation has been designed in the A^*A inner-product, although the normal equations are never formed. This is necessary because indefinite and non-Hermitian A do not induce inner-products, as in the Hermitian case. A^*A is used instead of AA^* so that user-provided near null-space modes are preserved—i.e., $AB \approx \mathbf{0}$ implies $A^*AB \approx \mathbf{0}$. Additionally, this energy principle easily extends to complex-valued systems.

3.3. *F/C*-based energy minimization.

In this section, we outline a strategy for adapting the SA-based scheme to a root-node-based scheme which has a direct correspondence to classical F/C-type AMG. Specifically, root-node SA refers to viewing each standard SA aggregate as being centered around a root node and considering a tentative prolongator that injects root nodes from the coarse grid to the fine grid. This is constructed by taking a standard tentative prolongator (obtained via (3.3)), scaling each column so that the entry corresponding to a root-node row is one, and then zeroing out all other nonzero entries in root-node rows. To preserve interpolation of the near null-space, the coarse near null-space representation must be scaled accordingly and now corresponds to injection of the fine near null-space. The sparsity pattern is also identical to the SA-based scheme with the exception of root-node rows, which correspond to injection, thereby giving the F/C-style sparsity pattern in (3.1). With an F/C sparsity pattern, an initial feasible $P^{(0)}$, and a coarse near null-space representation, energy minimization can now be applied.

This approach has several attractive properties. As discussed in the next subsection, the identity in the lower block of (3.1) guarantees that the columns of P are linearly independent. The restriction of the sparsity pattern corresponding to root nodes effectively reduces the space in which energy is minimized. In particular, the energy at the root nodes or C-points does not play a strong role in the minimization process, and this is natural from a Schur complement perspective. As discussed in [11], ideal weights for P_F correspond to

$$P_{ideal} = \begin{bmatrix} -A_{FF}^{-1}A_{FC} \\ I \end{bmatrix}. \tag{3.4}$$

The submatrices of A in (3.4) are defined by the list of indices C, which simply lists the root nodes or C-points, and by the list of indices $F = \{1, \ldots, n\} \setminus C$. Intuitively, P_{ideal} has "zero" energy at F-points because AP_{ideal} is zero at all F-points. However, the energy at C-points is nonzero and not relevant for this ideal prolongator.

One drawback arises when the number of near null-space modes exceeds the number of degrees-of-freedom per node[2] on the fine level. The QR procedure described earlier effectively sets the number of degrees-of-freedom per node on the coarse level so that it equals the number of near null-space modes. This means, however, that there is not a one-to-one correspondence between coarse and fine level degrees-of-freedom

[2] A common SA approach when multiple variables exist at each spatial location for the original matrix problem is supernodes [26]. Supernodes cause aggregation to happen in two stages at the finest level. First, supernodes are formed by aggregating degrees-of-freedom at the same spatial location together. Second, strength-of-connection information is used to aggregate the supernodes.

and so it is not possible to simply normalize columns so that a root-node entry has a value of one. This is because some columns do not correspond to a unique root-node entry. There are several ways of addressing this. One approach is to examine each aggregate and define a set of degrees-of-freedom equal to the number of near null-space modes such that an identity (i.e., injection) operator can be defined for this aggregate. Once this set (which may include some non-root nodes) is defined, the procedure of normalizing columns and zeroing remaining entries within a row proceeds as before. There are other possible extensions, but these are not pursued in this paper.

3.4. Conditioning of prolongator columns. Recursive application of the multigrid idea imposes an additional requirement on grid transfer operators. Specifically, relaxation methods must be applied to the coarse level matrix, which is defined by the triple matrix product RAP. Therefore, it is essential that R and P do not introduce a poorly conditioned coarse level operator that would render relaxation less effective. For example, a P with linearly dependent columns gives rise to a singular coarse level operator, even if the fine level operator is nonsingular (or even well-conditioned). More generally, poorly conditioned prolongator columns cause the conditioning of coarse level operators to suffer. Of course, the $P^{(0)}$ defined by (3.1) cannot have linearly dependent columns due to the identity associated with C-points. This also holds for $P^{(i)}$ if $\mathcal{N}$ is chosen so that this identity is maintained throughout the minimization.[3] Thus, conditioning of the grid transfers for F/C-based algorithms is normally not a concern.

However, linear dependence is possible with SA-based energy minimization. In particular, two prolongator columns have an identical minimum energy solution if the sparsity pattern and near null-space constraints are identical. Recall that the sparsity pattern is governed by $|S|^d|P^{(0)}|$ and so two columns with an identical sparsity pattern is most likely to occur when $P^{(0)}$ has two columns with an identical sparsity pattern. This happens for columns lying inside the same block column of $P^{(0)}$ (i.e., the case where $m > 1$). One way to address near linear dependence in prolongator columns is to postprocess with a local QR step to ensure that columns associated with the same aggregate are orthogonal. The Q replaces the nonzeros in the block column of P, and B^C is left multiplied by R at the degrees-of-freedom corresponding to that block column. Multiplying B^C with R maintains $PB^C = B$. An alternative and potentially computationally less expensive approach is to ensure distinct sparsity patterns inside each block column of P. In practice, however, since only a few Krylov iterations are taken, near linear dependence of the prolongation columns is typically not observed.

4. Implementation and computational cost. Implementation of Krylov minimization algorithms requires dot products involving Krylov vectors, matrix-vector products involving $I_q \otimes A$, and application of Q. In this section, we discuss these kernels and comment on the computational cost associated with energy minimization. We also detail the QA-GMRES and QAQ-CG algorithms and describe how to efficiently solve the formal problem (2.2) with Krylov methods by avoiding the formation of $(I_q \otimes A)$, $\Delta\check{P}$, and $\check{P}^{(0)}$. Using the available sparse matrices A, ΔP, and $P^{(0)}$, we define matrix-vector multiplication with $(I_q \otimes A)$, in addition to Euclidean inner-products and norms on vectors formed from sparse matrices with the $\check{}$ expansion.

[3] Prolongator basis functions with minimum energy typically decay *smoothly* to zero from the value of one at the C-points. That is, entries of P_F are typically smaller than one, and so orthogonality of the identity block implies that prolongator columns are *relatively far* from linearly dependent.

Matrix-vector multiplication is implemented using

$$Y = AP, \quad \text{where } (I_q \otimes A)\check{P} = \check{Y}. \tag{4.1}$$

The Euclidean inner-product is implemented using the identity

$$\langle \check{P}, \check{P}^{(0)} \rangle_2 = \langle P, P^{(0)} \rangle_F, \tag{4.2}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner-product. This inner-product computes an elementwise conjugation on the first argument, followed by an elementwise matrix-matrix multiplication. The resulting matrix is then summed elementwise. The Euclidean norm of $\check{P}$ is implemented as the Frobenius norm

$$\left\| \check{P} \right\|_2 = \left\| P \right\|_F. \tag{4.3}$$

The final key computational kernel involves application of Q, which includes inversion of XX^*. Using the constraint $PB_j^C = B_j$ with $1 \le j \le m$, the ith row of P yields m rows in X (one for each value of j). Only these m rows of X constrain the ith row of P, implying that XX^* can be permuted to block diagonal form with blocks of size $m \times m$ (see section 2.1.1). Thus, application of Q requires a factorization of a series of $m \times m$ matrices, discussed in the next paragraph. In a typical case where $m = 1$ and both B and B^C are the constant function, application of Q computes the average of the nonzeros values in each row of ΔP and subtracts this from each nonzero in that row. As a result, the row sums of ΔP are zero.

More generally, we preserve the near null-space (see `project_out`(ΔP)) by using a rowwise projection, which results in $X \Delta \check{P} = \mathbf{0}$. We right-multiply the nonzero part of the ith row of ΔP with $(I - \hat{B^C}((\hat{B^C})^* \hat{B^C})^\dagger (\hat{B^C})^*)$. $\hat{B^C}$ is B^C with rows restricted to the specified sparsity pattern for row i of ΔP, and corresponds to a diagonal block in the permuted X from Section 2.1.1. As mentioned in section 2.1.1, possible rank deficiencies require use of the Moore–Penrose pseudoinverse †, so that Q is uniquely defined. Additionally for efficiency, this small $(m \times m)$ pseudoinverse of $((\hat{B^C})^* \hat{B^C})$ needs to be computed only once for each row, since the nonzero pattern is preset.

In the algorithms that follow (see `enforce()`), we enforce the sparsity pattern constraint by using the sparsity structure of $|S|^d |P^{(0)}|$, where $d \in \{0, 1, 2, \dots\}$. Applying $| \cdot |$ ensures that the interpolation stencil grows as expected. A more efficient option than zeroing out entries, which we implemented for our experiments in the package PyAMG [1], is to compute values during prolongation smoothing only at the desired sparsity pattern.

Remark 4.1. Using $|S|^d |P^{(0)}|$ to generate interpolation sparsity patterns is a generalization of the classic SA technique of weighted-Jacobi prolongation smoothing with the filtered matrix of A. Both approaches desirably expand interpolation stencils only in the direction of strong connections.

There are additional computational savings in the case of complex-symmetric A. As shown in [18], the QA^*AQ-CGNR prolongation smoother satisfies the property that $R = P^T$ (i.e., the strict non-Hermitian transpose of P) is an appropriate choice for R when A is complex-symmetric. Similar to [18], the QA-GMRES algorithm can be shown to yield appropriate $R = P^T$ for complex-symmetric A. Overall, the complex-symmetry of A is reflected in the choice of $R = P^T$ when complex conjugation distributes over the prolongation smoothing algorithm [15].

4.1. GMRES prolongation algorithm. All the prolongation smoothers use diagonal preconditioning, as represented by D^{-1}. For systems where supernodes are

```
Algorithm 2: gmres_prolongation_smoother($A$, $P^{(0)}$, $B^C$, $S$, $d$, $iter$)
$\mathcal{N} = |P^{(0)}|$
for $i = 1$ to $d$ do
    $\mathcal{N} = |S|\mathcal{N}$

$D =$ diag($A$)                                      {Diagonal preconditioner}
$R = -D^{-1}AP^{(0)}$                                     {Initial residual}
$R =$ enforce($R$,$\mathcal{N}$)                               {Enforce sparsity on $R$}
$R =$ project_out($R, B^C$)                               {Enforce $RB^C = \mathbf{0}$}
$\beta = \|R\|_F$
$V^{(0)} = R/\beta$

for $i$ to $iter$ do
    $V^{(i)} = D^{-1}AV$                                  {New search direction}
    $V^{(i)} =$ enforce($V^{(i)}$,$\mathcal{N}$)                  {Enforce sparsity on $V^{(i)}$}
    $V^{(i)} =$ project_out($V^{(i)}, B^C$)                  {Enforce $V^{(i)}B^C = \mathbf{0}$}

    for $j = 1$ to $i$ do                                         {Modified GS}
        $H_{j,i} = \langle V^{(j)}, V^{(i)} \rangle_F$
        $V^{(i)} = V^{(i)} - H_{j,i}V^{(j)}$
    $H_{i+1,i} = \left\|V^{(i)}\right\|_F$

$y =$ solve($H, \beta e_1$)              {Best update to $P^{(0)}$ in Krylov space}
for $j = 1$ to $i$ do                   {use $i + 1 \times i$ upper-left block of $H$}
    $P^{(0)} = P^{(0)} + y_j V^{(j-1)}$

return $P^{(0)}$
```

used, D^{-1} is a block inverse where the blocksize is the supernode size. Following the motivation earlier in this section, the proposed Krylov methods operate on sparse matrices with the Frobenius norm and inner-product. This yields the desired equivalence with the Euclidean inner-product and norm.

Algorithm 2 executes GMRES in Krylov space (2.20d). Equation (2.20a) is solved with $\Delta\check{P}^{(0)} = \mathbf{0}$ subject to the constraints enforced by `enforce()` and `project_out()`. As presented, Algorithm 2 iterates a fixed number of times, but could be changed to halt with a residual tolerance.

4.2. CG prolongation algorithm. Algorithm 3 executes QAQ-CG in Krylov space (2.25c). Equation (2.25a) is solved with $\Delta\check{P}^{(0)} = \mathbf{0}$ subject to the constraints enforced by `enforce`() and `project_out`().

Algorithm 3 is easily extended to the QA^*AQ-CGNR framework by replacing multiplications by A with multiplications by A^*A. A^*AP is explicitly formed, as opposed to forming only AP as in standard CGNR, so that the constraints are applied to each new search direction.

5. Numerical results. In this section we consider several problem types, both Hermitian and non-Hermitian, that support the energy minimization approach. Unless otherwise noted, 4 iterations of the energy minimization process are taken, i.e.,

Algorithm 3: cg_prolongation_smoother(A, $P^{(0)}$, B^C, S, d, $iter$)

```
$\mathcal{N} = |P^{(0)}|$
for $i = 1$ to $d$ do
    $\mathcal{N} = |S|\mathcal{N}$

$D =$ diag$(A)$                                  {Diagonal preconditioner}
$R = -AP^{(0)}$                                  {Initial residual}
$R =$ enforce$(R, \mathcal{N})$                  {Enforce sparsity on $R$}
$R =$ project_out$(R, B^C)$                      {Enforce $RB^C = \mathbf{0}$}

for $i$ to $iter$ do
    $Z = D^{-1}R$
    $\gamma = \langle R, Z \rangle_F$

    if $i$ is 1 then
        $Y = Z$
    else
        $\beta = \gamma/\gamma_{old}$; $Y = Z + \beta Y$       {New search direction}

    $\gamma_{old} = \gamma$
    $Y_A = AY$
    $Y_A =$ enforce$(Y_A, \mathcal{N})$          {Enforce sparsity on $Y_A$}
    $Y_A =$ project_out$(Y_A, B^C)$              {Enforce $Y_A B^C = \mathbf{0}$}
    $\alpha = \gamma/\langle Y, Y_A \rangle_F$
    $P^{(i)} = P^{(i-1)} + \alpha Y$             {Update prolongator}
    $R = R - \alpha Y_A$                         {Update residual}

return $P^{(i)}$
```

$iter = 4$. Given the flexibility of this approach, we consider problems in both a root-node and classic SA context. The results are similar, exhibiting convergence and scalability for a number of challenging applications where other solvers typically do not.

5.1. Root-node style experiments.

Non-Hermitian problem. We begin with a model problem of one-dimensional convection-diffusion,

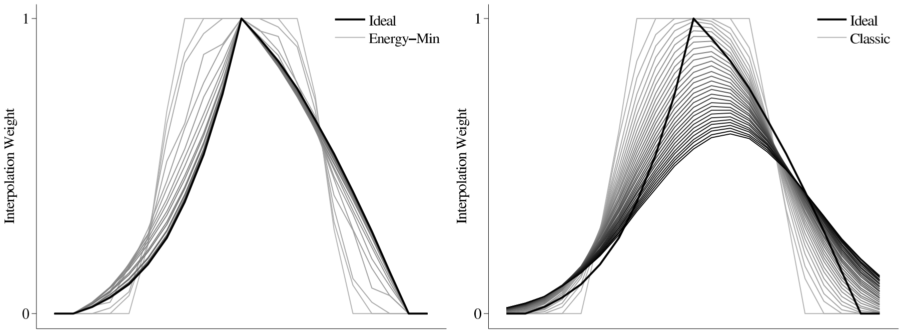
$$-u_{xx} + c(x)u_x = f, \tag{5.1}$$

with periodic boundary conditions. The PDE is discretized with central differences to obtain a nonsymmetric operator with stencil,

$$\begin{bmatrix} -\frac{1}{h^2} + \frac{c}{2h} & \frac{2}{h^2} & -\frac{1}{h^2} - \frac{c}{2h} \end{bmatrix}, \tag{5.2}$$

on a uniform grid over $[0, 1]$. A choice of $c = 2/h$ yields pure upwinding.

Importantly, $-A_{FF}^{-1}A_{FC}$ is sparse here, thus allowing for a direct comparison between prolongation basis functions from P_{ideal} with the basis functions from QA-GMRES. Since we are examining root-node style SA, the tentative prolongator is



(a) Energy minimization. (b) Classic.

FIG. 5.1. *Classical and energy minimized SA local prolongation basis functions.*

constructed according to (3.1) with the P_F block based on a B equal to a constant function.

We depict the basis function—i.e., column of P—in Figure 5.1 for the aggregate with root node located in the center of the domain. Since the basis function is limited to the nonzero pattern from P_{ideal}, only the local nonzero portion of each basis function is depicted, in addition to several extra degrees-of-freedom on the right and left to indicate that values are zero outside of the range of the plot.

Figure 5.1(a) shows how the basis function evolves over 32 iterations of QA-GMRES for $c(x) = \sin(2\pi x)$. Here, $c(x)$ completes one period over the domain. Importantly, the figure shows that the sparsity pattern constraint allows QA-GMRES to take many iterations and not grow the stencil outside of the allowed pattern. Eventually, QA-GMRES converges to P_{ideal}. Although not depicted, it is worth noting that only a few QA-GMRES iterations are needed before there is no additional gain from an AMG convergence perspective.

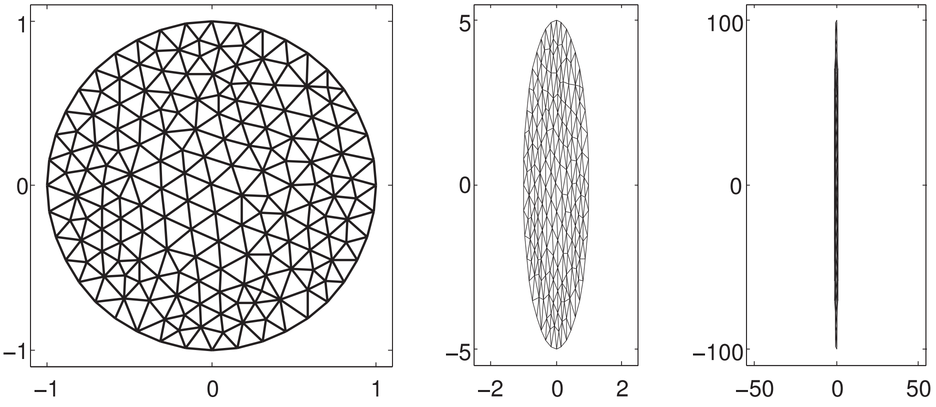
Figure 5.1(b) shows the result of using classic Jacobi-based prolongation smoothing. Again, 32 iterations are depicted in order to convey the limiting behavior of classic prolongation smoothing; although this expands the stencil significantly (beyond 4 iterations), we clip the figure for comparison. Classical prolongation smoothing does not yield basis functions that approximate P_{ideal} well. The profile of the ideal basis function is not recovered in the process. Typically, A_{FF}^{-1} is not sparse; so, in general, energy minimization attempts to approximate P_{ideal}.

Hermitian problem. The Hermitian test problem is a two-dimensional diffusion problem constructed to be irregular and particularly difficult. We solve the classic two-dimensional elliptic equation,

$$u_{xx} + u_{yy} = f, \tag{5.3}$$

with homogeneous Dirichlet boundary conditions and discretize with linear simplicial elements. The difficulty is due to the domain, which is an unstructured mesh of a stretched circle centered at the origin. The stretching factor is only in the y-direction and yields very thin elements in the mesh, resulting in a matrix that is highly anisotropic with highly irregular rows. The stretching of various meshes is depicted in Figure 5.2. For our experiments, a factor of 100 is used.

Due to the inherent difficulty of this problem and also in order to illustrate the generality of the proposed methods, we couple QAQ-CG to compatible relaxation



(a) Stretching = 1. (b) Stretching = 5. (c) Stretching = 100.

FIG. 5.2. *Stretched circular meshes.*

(CR) [14]. CR chooses an appropriate F/C-splitting and the C-points become the root nodes of aggregates. The matrix stencil is then used to find a nonoverlapping sparsity pattern $\mathcal{N}$ for $P^{(0)}$. We use the constant vector for B and the pre- and postsmoother is an overlapping Schwarz method, which was found to be critical for achieving acceptable performance.

In Table 5.1, results for CG preconditioned with V(1,1)-cycles are presented, and we compare classic Jacobi prolongation smoothing to QAQ-CG. Solvers with 2 and 3 levels were generated with moderate operator complexities that were approximately 1.75 for the 2-level method and 2.1 for the 3-level method. Even for these modest problem sizes, the use of the proposed prolongation smoother has an immediate positive impact on performance.

TABLE 5.1
Preconditioned CG iterations.

		Prolongation smoother	
	n	Jacobi	QAQ-CG
Levels=2	13 373	24	14
	53 069	42	18
Levels=3	13 373	24	14
	53 069	42	19

5.2. Classical SA experiments. In this section, we consider numerical experiments that use the classic (i.e., non-root-node) approach to SA.

Hermitian problems. We expect the proposed methods to also add robustness to standard multigrid test problems. Consider the two-dimensional anisotropic diffusion PDE

$$-(c^2 + \epsilon s^2)u_{xx} - 2(1-\epsilon)cs\, u_{xy} - (\epsilon c^2 + s^2)u_{yy} = f, \tag{5.4}$$

where $\epsilon = 0.001$, $c = \cos(\Theta)$, $s = \sin(\Theta)$, and Θ is the angle of rotation. We examine vertical anisotropy for $\Theta = 0$ and two more difficult rotated anisotropies. We use Q1 finite elements to discretize the problem on the unit box with a regular grid and Dirichlet boundary conditions.

We use Algorithm 1 to set up the SA hierarchy. V(1,1)-cycles precondition CG to a tolerance of 10^{-8}. Gauss–Seidel is used for the pre- and postsmoother and to presmooth B on each level, where the initial B is the standard constant used for diffusion problems.

We compare various combinations of strength-of-connection measures and prolongation smoothers. Classical Jacobi prolongation smoothing is compared with QAQ-CG, which iterates 4 times with an interpolation stencil width of 2. The coarse grid choice is very influential, so we show results for both the classic strength-of-connection measure with an appropriate drop tolerance for anisotropic problems of 0.25 and the evolution measure [19] with its standard drop tolerance of 4.0. Operator complexities remained low for all tests.

The number of preconditioned CG iterations is presented in Table 5.2. The combination of the evolution measure and QAQ-CG produces the most robust and mesh-independent solver. The beneficial effect of the proposed method is observed by comparing the "Jacobi" column with the "QAQ-CG" column for either strength-of-connection measure. Overall, it is clear that the quality strength-of-connection decisions provided by the evolution measure are needed for the best performance with QAQ-CG .

Table 5.2
Preconditioned CG iterations for two-dimensional anisotropic diffusion.

	Prol. smoother:	Jacobi		QAQ-CG	
	Strength:	Classical	Evol.	Classical	Evol.
Vertical anisotropy	$h = 1/16$	9	9	8	8
	1/32	16	10	11	8
	1/64	29	11	18	10
	1/128	57	11	33	11
Rot. by $\pi/4$ anisotropy	1/16	7	8	8	8
	1/32	14	15	12	13
	1/64	29	23	18	14
	1/128	58	24	31	14
Rot. by $\pi/8$ anisotropy	1/16	12	12	10	10
	1/32	19	18	12	11
	1/64	34	26	20	14
	1/128	65	36	34	18

Next, we explore QAQ-CG in the context of multiple near null-space modes. We examine an isotropic linearized elasticity problem where $B \in \mathbb{R}^{n,6}$ and is equal to the six rigid body modes. Linearized elasticity is defined by

$$-\text{div}\left(\lambda \text{ tr}\left(\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)/2\right) I + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)\right) = f, \tag{5.5}$$

where λ and μ are the Lamé parameters, I is the identity matrix, and tr() is the trace function. The GetFem++ package [20] is used to discretize the problem with tetrahedrons for a steel tripod and a downward external force applied to the top of the tripod. An example coarse mesh is given in Figure 5.3. Table 5.3 (cf. [24]) presents the resulting preconditioned CG iterations. The use of prerelaxation for the near null-space modes ($\eta = 5$) is an important factor in achieving the results, especially for weighted-Jacobi prolongation smoothing. QAQ-CG in conjunction with evolution measure yields the most robust performance.

Non-Hermitian problems. We examine a convection-diffusion problem for recirculating flow and the classic lid-driven cavity problem. Both result in nonsymmetric matrix problems that challenge multigrid. The lid-driven cavity is discretized with a marker and cell (MAC) approach, and we consider solving the $(1, 1)$ block of the

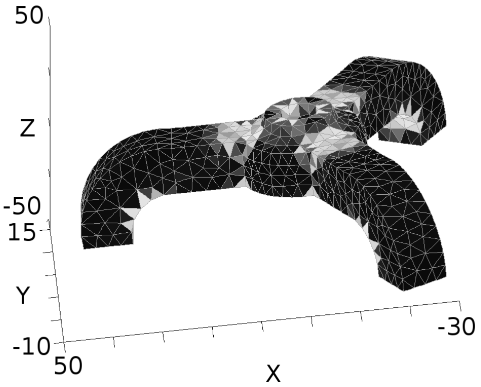


FIG. 5.3. *Example coarse tripod mesh.*

system as part of a Schur-complement approach. Sample streamlines are depicted in Figure 5.4(a).

TABLE 5.3
Preconditioned CG iterations, linearized elasticity.

	Prol. smoother:	Jacobi		QAQ-CG	
	Strength:	Classical	Evol.	Classical	Evol.
$p = 1$	$n =$ 2 757	16	19	11	12
	16 341	29	21	12	12
	109 551	41	34	14	13
$p = 2$	16 341	58	34	24	15
	109 551	82	34	25	17

The recirculating flow problem is

$$-\epsilon \Delta u + \mathbf{b} \cdot \nabla u = f, \quad \text{with} \tag{5.6}$$

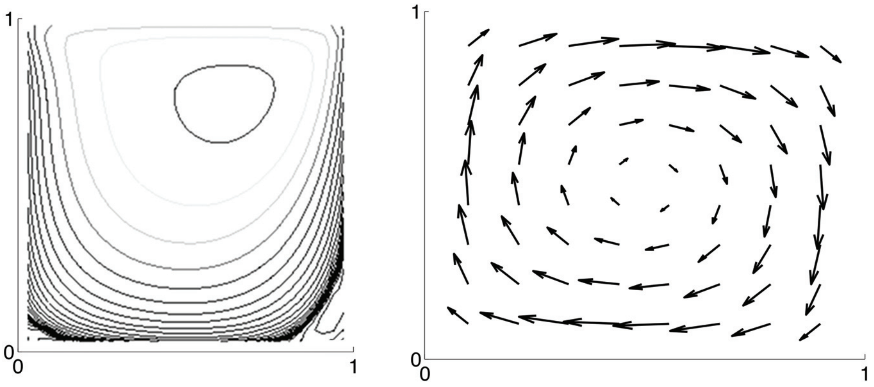
$$\mathbf{b} = \begin{bmatrix} 4x(x-1)(1-2y) \\ -4y(y-1)(1-2x) \end{bmatrix}, \tag{5.7}$$

and Dirichlet boundary conditions on the unit box. A plot of $\mathbf{b}()$ is given in Figure 5.4(b).

We use Algorithm 1 to set up the SA hierarchy and use V(1,1)-cycles to precondition GMRES to a tolerance of 10^{-8}. Since the matrix is nonsymmetric, we use Gauss–Seidel normal residual (NR) for the pre- and postsmoother and to presmooth B^L and B^R. Gauss–Seidel NR is essentially Gauss–Seidel on the normal equations and satisfies the A^*A energy principle. The initial basis for the left and right near null-spaces, B^L and B^R, is the constant function.

We compare various combinations of strength-of-connection measures and prolongation smoothers. Classical Jacobi prolongation smoothing is compared with QA-GMRES, which iterates 4 times with an interpolation stencil width of 1. Importantly, R is smoothed by A^* in both cases. The coarse grid choice is influential, so we also compare the classic strength-of-connection measure with an appropriate $\theta = 0.0$ against the evolution measure with $\theta = 8.0$. Tuning θ for the evolution measure does not have a dramatic impact on performance, but $\theta = 8.0$ was superior to $\theta = 4.0$. In all test cases, the operator complexity remained low.

The third method compared is labeled "partial-SA," which refers to partially smoothed aggregation. Here, R is never smoothed and there is no presmoothing of the



(a) Lid-driven cavity.

(b) Recirculating flow.

FIG. 5.4. *Sample streamlines (lid-driven cavity) and recirculating flow field (***b***).*

TABLE 5.4
Preconditioned GMRES iterations for the lid-driven cavity.

	Prol. smoother:	Jacobi		part-SA		QA-GMRES	
	Strength:	Classical	Evol.	Classical	Evol.	Classical	Evol.
Re = 100	$h = 1/64$	14	12	17	15	12	12
	1/128	13	13	17	21	12	12
	1/256	15	13	17	29	14	13
Re = 1000	1/64	35	18	43	31	33	19
	1/128	36	50	42	78	38	20
	1/256	75	42	50	95	41	18

TABLE 5.5
Preconditioned GMRES iterations for recirculating flow.

	Prol. smoother:	Jacobi		part-SA		QA-GMRES	
	Strength:	Classical	Evol.	Classical	Evol.	Classical	Evol.
$\epsilon = 10^{-4}$	$h = 1/64$	137	73	54	33	52	23
	1/128	151+	136	83	90	101	24
	1/256	151+	151+	90	151+	151+	25
	1/512	151+	151+	112	151+	151+	27
$\epsilon = 10^{-5}$	1/64	150	93	63	39	61	24
	1/128	151+	151+	127	81	151+	28
	1/256	151+	151+	151+	151+	151+	33
	1/512	151+	151+	151+	151+	151+	33

vectors B^L and B^R, so that partially smoothed aggregation is a point of comparison to previous nonsymmetric approaches to SA.

The GMRES iterations are detailed in Tables 5.4 and 5.5. The combination of the evolution measure and QA-GMRES produces the most robust and mesh-independent solver in both cases. Overall, it is clear that quality strength-of-connection decisions are needed for QA-GMRES to succeed. With the quality strength-of-connection information provided by the evolution measure, the "Jacobi" column is compared with the "QA-GMRES" column; again we observe the value of using the proposed prolongation smoother.

6. Conclusion. There is a need for a general iterative process in both SA-based and F/C-based AMG for improving an initial prolongation operator P. The iterative process should allow for indefinite or non-Hermitian A by improving R and P with a meaningful definition of energy. During the process, it is critical to control complexity through restrictions on the sparsity pattern, and to incorporate a priori knowledge about near null-space modes into interpolation. An approach to improving grid transfers has been proposed based on Krylov methods where the Krylov space is restricted to satisfy a set of constraints. As demonstrated by the numerical experiments, the proposed approach fills this need for both SA- and F/C-style AMG.

For non-Hermitian or indefinite problems, QA^*AQ-CGNR and QA-GMRES are competitive approaches. There are several considerations, however. The conditioning of QA^*AQ-CGNR may detract from the method in some cases and the normal equations often hide physical artifacts such as convection. Additionally, new search directions in the algorithm require multiplication by A^*A, which is more expensive than multiplication by A. In comparison, QA-GMRES does not suffer from these issues. On the other hand, the QA^*AQ-CGNR approach is able to utilize a 3-term recurrence and minimizes each column of P in the A^*A-norm, while QA-GMRES minimizes P globally in the related $Q_A^*Q_A$-seminorm and has no advantageous recurrence.

For HPD matrices, the QAQ-CG approach has a clear advantage. This approach enjoys a 3-term recurrence and minimizes each column of P in the A-norm.

One drawback of the proposed methods is the computational and memory costs. For the two CG-based methods, three vectors—i.e., matrices with the preset sparsity pattern—must be stored. For QA-GMRES, the number of iterations defines the number of "vectors" that must be stored. Computationally, the proposed methods are more costly than standard prolongation smoothing because each minimization step includes calculations that are on the order of classical SA prolongation smoothing, i.e., one application of weighted-Jacobi to $P^{(0)}$. In [18], the CGNR-based approach became the most expensive part of the SA setup phase, but nonetheless remained commensurate to the system solve time. However, only a few iterations are typically used in our energy minimization approach, resulting in a low operational cost.

In comparison to other methods, there are several advantages of the proposed energy minimization algorithms. First, the framework allows direct control of the sparsity (and cost) of the system, regardless of the number of prolongation smoothing steps taken. Moreover, long-distance interpolation is straightforward by growing interpolation stencils in the direction of strong connections. Classical SA provides only fixed high-degree matrix polynomial prolongation smoothers for long-distance interpolation and provides little control over the sparsity pattern. Entries in A are dropped before prolongation smoothing, but complexity grows quickly for more than one iteration. Classical F/C AMG is similarly rigid with respect to long-distance interpolation and sparsity patterns, but specialized formulas do exist for long-distance interpolation. Finally, a benefit of the constraints is that they are straightforward to integrate into a minimization process, and they rely on established iterative methods for improving interpolation weights. Neither classic SA nor F/C AMG provides similar features, and neither targets general classes of matrices.

Compared to other advanced approaches for calculating P, the methods discussed in section 2 offer some advantages. The adaptive nonsymmetric SA approach [8] and the restriction smoothing method of local damping parameters [23] share basic algorithmic properties of the classic SA approach and hence share disadvantages. Long-distance interpolation and multiple smoothing iterations for P are not available without large increases in complexity. Additionally, these methods are not equipped to handle a general collection of constraints. Compared to [16], we use constrained Krylov methods to yield a robust minimization in an energy-like norm, as opposed to constrained steepest descent. Allowing for different Krylov methods enhances robustness by addressing indefinite or non-Hermitian problems. Additionally, treating constraints near boundaries through presmoothing B is more automatic than [16], where boundary nodes are an input and receive special treatment while smoothing P.

Acknowledgment. The software package PyAMG [1] was instrumental to our numerical tests and some energy-minimization algorithms are publicly available in this package.

REFERENCES

[1] W. N. Bell, L. N. Olson, and J. Schroder, *PyAMG: Algebraic multigrid solvers in Python v*1.0, Release 1.0, 2008; available online from http://www.pyamg.org.

[2] A. Brandt, *Multigrid Techniques:* 1984 *Guide with Applications to Fluid Dynamics*, Tech. Report 85, GMD-Studie, Sankt Augustin, Germany, 1984.

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

[4] A. Brandt, *Multiscale scientific computation: Review* 2001, in Multiscale and Multiresolution Methods, Springer-Verlag, Berlin, 2002, pp. 3–95.

[5] A. Brandt, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid (AMG) for sparse matrix equations*, in Sparsity and Its Applications, D. J. Evans, ed., Cambridge University Press, Cambridge, UK, 1984, pp. 257–284.

[6] J. Brannick and L. Zikatanov, *Algebraic multigrid methods based on compatible relaxation and energy minimization*, in Domain Decomposition Methods in Science and Engineering XVI, O. Widlund and D. E. Keyes, eds., Lect. Notes Comput. Sci. Eng. 55, Springer-Verlag, Berlin, 2007, pp. 15–26.

[7] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation* (αSA) *multigrid*, SIAM Rev., 47 (2005), pp. 317–346.

[8] M. Brezina, T. Manteuffel, S. McCormick, J. Ruge, and G. Sanders, *Towards adaptive smoothed aggregation (αSA) for nonsymmetric problems*, SIAM J. Sci. Comput., 32 (2010), pp. 14–39.

[9] G. Carre, G. Carte, H. Guillard, and S. Lanteri, *Multigrid strategies for CFD problems on non-structured meshes*, in Multigrid Methods VI, Springer-Verlag, Ghent, 2000, pp. 1–10.

[10] J. Dendy, *Black box multigrid for nonsymmetric problems*, Appl. Math. Comput., 13 (1983), pp. 261–283.

[11] R. D. Falgout and P. S. Vassilevski, *On generalizing the algebraic multigrid framework*, SIAM J. Numer. Anal., 42 (2004), pp. 1669–1693.

[12] A. Janka, *Multigrid Methods for Compressible Laminar Flow*, Ph.D. thesis, University of Nice-Sophia Antipolis, 2002.

[13] T. V. Kolev and P. S. Vassilevski, *AMG by element agglomeration and constrained energy minimization interpolation*, Numer. Linear Algebra Appl., 13 (2006), pp. 771–788.

[14] O. E. Livne, *Coarsening by compatible relaxation*, Numer. Linear Algebra Appl., 11 (2004), pp. 205–227.

[15] S. P. MacLachlan and C. W. Oosterlee, *Algebraic multigrid solvers for complex-valued matrices*, SIAM J. Sci. Comput., 30 (2008), pp. 1548–1571.

[16] J. Mandel, M. Brezina, and P. Vaněk, *Energy optimization of algebraic multigrid bases*, Computing, 62 (1999), pp. 205–228.

[17] S. F. McCormick and J. W. Ruge, *Multigrid methods for variational problems*, SIAM J. Numer. Anal., 19 (1982), pp. 924–929.

[18] L. N. Olson and J. B. Schroder, *Smoothed aggregation for Helmholtz problems*, Numer. Linear Algebra Appl., 17 (2010), pp. 361–386.

[19] L. N. Olson, J. B. Schroder, and R. S. Tuminaro, *A new perspective on strength measures in algebraic multigrid*, Numer. Linear Algebra Appl., 17 (2010), pp. 713–733.

[20] Y. Renard, *The GetFem++ Project*, http://download.gna.org/getfem/html/homepage/.

[21] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, S. F. McCormick, ed., Frontiers Appl. Math. 3, SIAM, Philadelphia, 1987, pp. 73–130.

[22] M. Sala, *An algebraic 2-level domain decomposition preconditioner with applications to the compressible Euler equations*, Internat. J. Numer. Methods Fluids, 40 (2002), pp. 1551–1560.

[23] 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.

[24] J. B. Schroder, *Generalizing Smoothed Aggregation-Based Algebraic Multigrid*, Ph.D. thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2010.

[25] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.

[26] P. Vaněk, M. Brezina, and J. Mandel, *Convergence of algebraic multigrid based on smoothed aggregation*, Numer. Math., 88 (2001), pp. 559–579.

[27] P. Vaněk, A. Janka, and H. Guillard, *Convergence of Algebraic Multigrid Based on Smoothed Aggregation* II: *Extension to a Petrov-Galerkin Method*, Tech. Report 3683, INRIA, 1999.

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

[29] P. S. Vassilevski, *General constrained energy minimization interpolation mappings for AMG*, SIAM J. Sci. Comput., 32 (2010), pp. 1–13.

[30] C. Wagner, *On the algebraic construction of multilevel transfer operators*, Computing, 65 (2000), pp. 73–95.

[31] W. L. Wan, T. F. Chan, and B. Smith, *An energy-minimizing interpolation for robust multigrid methods*, SIAM J. Sci. Comput., 21 (2000), pp. 1632–1649.

[32] J. Xu and L. Zikatanov, *On an energy minimizing basis for algebraic multigrid methods*, Comput. Vis. Sci., 7 (2004), pp. 121–127.