SIAM J. Sci. Comput.
Vol. 30, No. 3, pp. 1548–1571

ALGEBRAIC MULTIGRID SOLVERS FOR COMPLEX-VALUED MATRICES*

SCOTT P. MACLACHLAN[†] AND CORNELIS W. OOSTERLEE[‡]

Abstract. In the mathematical modeling of real-life applications, systems of equations with complex coefficients often arise. While many techniques of numerical linear algebra, e.g., Krylov-subspace methods, extend directly to the case of complex-valued matrices, some of the most effective preconditioning techniques and linear solvers are limited to the real-valued case. Here, we consider the extension of the popular algebraic multigrid method to such complex-valued systems. The choices for this generalization are motivated by classical multigrid considerations, evaluated with the tools of local Fourier analysis, and verified on a selection of problems related to real-life applications.

Key words. multigrid, algebraic multigrid, complex-valued matrices

AMS subject classifications. 65F10, 65N22, 65N55

DOI. 10.1137/070687232

1. Introduction.

Many real-world physical systems may be modeled mathematically using the tools of partial differential equations. For many such models, the degrees of freedom are naturally real-valued, for example, displacements in an elastic body or velocities of a fluid. For some models, however, complex-valued degrees of freedom also arise naturally, as in frequency-domain modeling of electromagnetic waves or other phenomena. Because of the many interesting real-valued models, development of the numerical linear algebra tools needed for the solution of discrete linear systems has focused on the real-valued case. While some of these techniques may be easily extended to the complex-valued case (e.g., GMRES and BiCGStab for general matrices, or conjugate gradients for complex Hermitian matrices), many require special consideration to generalize the appropriate principles to the complex-valued case. Here, we consider the generalization of the algebraic multigrid method [6, 23], an effective solver (or preconditioner) for many linear systems that arise from the discretization of elliptic or parabolic differential equations.

The complex-valued linear systems considered here arise from different physical applications, for example, in modeling electromagnetic waves. Under the assumption of time-harmonic variation in the electromagnetic fields, Maxwell's equations may be reduced into a scalar Helmholtz equation with a complex shift (see, e.g., [16]). Similarly, when the acoustic (or elastic) wave equation is considered in the frequency domain, Sommerfeld boundary conditions and attenuation both introduce a complex component in the resulting Helmholtz equation; multigrid solvers for these (indefinite) matrices were considered in [15]. In the field of lattice quantum chromodynamics (QCD), a model of the interactions of fermions (or quarks) on a lattice is given in

[*] Received by the editors April 3, 2007; accepted for publication (in revised form) August 27, 2007; published electronically April 18, 2008. This research was supported by the European Community's Sixth Framework Programme, through Marie Curie International Incoming Fellowship MIF1-CT-2006-021927.

http://www.siam.org/journals/sisc/30-3/68723.html

[†] Department of Mathematics, Tufts University, 503 Boston Avenue, Medford MA 02155 (scott.maclachlan@tufts.edu).

[‡] Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Mekelweg 4, 2628 CD Delft, The Netherlands (c.w.oosterlee@tudelft.nl) and CWI, National Research Institute for Mathematics and Computer Science, Amsterdam, The Netherlands.

terms of a complex-valued gauge field that directly leads to a complex-valued linear system of equations [7, 24].

Multigrid methods are a family of techniques known to provide optimal (or near-optimal) solution of the linear systems that arise in many real-world applications. Through the careful coupling of a relaxation scheme (to reduce high-frequency errors) and a coarse-grid correction process (to reduce low-frequency errors), geometric multigrid techniques are among the most efficient solvers available for models with slowly varying coefficients [26]. For problems with significant heterogeneity, either in the coefficients of the continuous model or the geometry on which it is discretized, algebraic multigrid (AMG) techniques often perform as well as geometric techniques do for homogeneous models. Although first proposed in the 1980s [6, 23], there has been much recent interest in AMG because of its potential to handle large-scale models with realistic material properties and geometries, particularly in parallel environments [7, 16].

The success of AMG for a wide range of models is due to a careful combination of multigrid principles with more general ideas of numerical linear algebra. This combination, however, does not automatically yield a black-box approach for solving all types of linear systems. Effective multigrid performance results from a complementary choice of local relaxation and coarse-grid correction; AMG is not freed from these constraints, even though it is no longer dependent on knowledge of many details of the discrete problem under consideration. AMG performance for problems of structural mechanics, for example, is greatly improved if AMG is tempered with knowledge of the block structure of the linear system [23]. Such extensions to AMG are, in principle, straightforward and are not considered here.

While there has been much development of AMG for real-valued matrices, less investigation has occurred for complex-valued matrices. Lahaye et al. consider AMG for the Helmholtz equation with a complex shift and apply AMG to the real part of the matrix in order to define the coarse grids and interpolation operators [16]. For these models, the dominant part of the operator (corresponding to the second-order derivative terms) is entirely real, while the imaginary part represents only a mass matrix term, and, so, coarsening the complex-valued problem based on its real part is quite effective. Generalizing this approach, Reitzinger, Schreiber, and van Rienen propose using a real-valued auxiliary matrix to define the AMG hierarchy [21]. Such an approach is again appropriate when it is known that the dominant part of the operator may be represented by a real matrix. Both these approaches, however, require knowing how to split the given matrix in such a way as to define a real-valued auxiliary problem. Such an approach, then, is less general than the AMG approach for real-valued systems, which is based only on the entries in the linear system.

An alternate approach is to consider the equivalent real form of the complex system, splitting $A \in \mathbb{C}^{n\times n}$ into its real and complex parts, $A = A^{(R)} + \imath A^{(I)}$, and rewriting $A\mathbf{u} = \mathbf{b}$ as

$$\begin{bmatrix} A^{(R)} & -A^{(I)} \\ A^{(I)} & A^{(R)} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(R)} \\ \mathbf{u}^{(I)} \end{bmatrix} = \begin{bmatrix} \mathbf{b}^{(R)} \\ \mathbf{b}^{(I)} \end{bmatrix}.$$

Day and Heroux consider several possible orderings of the equivalent real form and show that ILU preconditioners applied to the equivalent real forms may be as effective as those applied to the complex formulation [13]. Adams uses an approach based on applying smoothed aggregation multigrid [28] to the equivalent real form [1]; such an approach was first considered in a two-level setting in [27]. The smoothed aggregation framework bases the multigrid interpolation operator on a specified set of so-called rigid body modes for the stiffness matrix (i.e., the dominant differential operator) and, so, these modes may be easily extended to match those of the equivalent real form. The adaptive smoothed aggregation multigrid method [9] is also applied to the equivalent real form of a system from QCD in [7].

Here, we consider the application of AMG directly to complex-valued linear systems, which may be more efficient than approaches based on the equivalent real form. In section 2, we give an introduction into the classical algebraic multigrid method, as it applies to symmetric real-valued matrices. Then, in section 3, we consider the extension of this algorithm to complex-valued matrices. These options are then analyzed using local Fourier analysis (LFA) in section 4. Finally, based on the choices recommended by the analysis in sections 3 and 4, a complex AMG algorithm is tested for several realistic models in section 5.

2. AMG for symmetric real-valued matrices.

Just as in all multigrid methods, the key to achieving efficiency in algebraic multigrid is an effective partitioning of the space of errors. In geometric multigrid, this partitioning is based on the ideas of smooth and oscillatory errors; those that appear smooth (relative to the underlying grid) are given to the coarse grid for resolution, while oscillatory errors must be appropriately attenuated by the chosen relaxation scheme. In algebraic multigrid, these roles are reversed; the subspace of errors that are effectively reduced by relaxation is taken to be fixed, and all complementary errors (the so-called algebraically smooth errors) must be reduced by an appropriate coarse-grid correction.

An important step in designing an effective AMG approach, then, is to characterize the errors that are slow to be attenuated by the chosen relaxation process. AMG was originally proposed as an extension of the successful geometric multigrid methods for finite-difference discretizations of Poisson's equation on irregular meshes [6]; as such, it is easily motivated by considering the performance of a simple relaxation scheme, such as the Jacobi iteration, for the class of M-matrices. A positive-definite (real-valued) matrix, A, is said to be an M-matrix if $a_{i,j} \leq 0$ for $i \neq j$. Furthermore, for an M-matrix, A, unknown i is said to strongly depend on unknown j if $-a_{ij} \geq \theta \max_{k\neq i}\{-a_{ik}\}$ for some $\theta \in (0, 1]$. Following these definitions, Jacobi and Gauss–Seidel relaxation can be shown to be slow to reduce errors that vary slowly between strongly connected nodes in the M-matrix, A, and that yield small residuals, $\mathbf{b} - A\mathbf{u}$, compared to the errors in $\mathbf{u}$ [4].

Consider, then, defining interpolation to a fine-grid node, i, for such an algebraically smooth error. Using the small-residual property, localized to node i, we write

$$(A\mathbf{e})_i = \sum_j a_{ij}e_j = a_{ii}e_i + \sum_{j\in F_i} a_{ij}e_j + \sum_{k\in C_i} a_{ik}e_k \approx 0, \tag{2.1}$$

where $\operatorname{adj}(i) = \{j : a_{ij} \neq 0\}$ is split into the two sets C_i and F_i, where C_i is the set of all coarse-grid points on which i strongly depends and $F_i = \operatorname{adj}(i) \setminus C_i$. Assuming equality in (2.1) gives

$$a_{ii}e_i = -\sum_{j\in F_i} a_{ij}e_j - \sum_{k\in C_i} a_{ik}e_k \tag{2.2}$$

so that, if the sum over F_i were not present, (2.2) could be used to directly define an interpolation stencil for node i in terms of its coarse-grid neighbors, $k \in C_i$. Thus, the task of defining interpolation is one of eliminating the connections to $j \in F_i$ from (2.2).

If a_{ij} is small, relative to other coefficients in row i of A, then e_j does not contribute much to this balance. To define "small," we return to the definition of strong connections; let

$$S_i = \left\{ j : -a_{ij} \geq \theta \max_{k \neq i} \{-a_{ik}\} \right\}.$$

Rather than completely removing connections to $F_i^w = F_i \setminus S_i$ from the balance, they are added to the diagonal by making the approximation that $e_j \approx e_i$. Note that this is a relatively safe choice; if point j has been wrongly classified as a weak connection, then $e_j \approx e_i$, since algebraically smooth errors vary slowly along strong connections. Thus, defining $F_i^s = F_i \cap S_i$ and $F_i^w = F_i \setminus F_i^s$, (2.2) is transformed into

$$\left(a_{ii} + \sum_{j \in F_i^w} a_{ij} \right) e_i = - \sum_{j \in F_i^s} a_{ij} e_j - \sum_{k \in C_i} a_{ik} e_k. \tag{2.3}$$

In the AMG coarse-grid selection process, each strongly connected fine-grid neighbor, j, of point i, is ensured to also be strongly connected to at least one point in C_i. Then, the value of e_j in (2.3) may be approximated by a weighted average of j's strongly connected neighbors in C_i. However, since a weak connection between j and $k \in C_i$ is reflected by a small coefficient, a_{jk}, it is safe to take a simpler approach and approximate e_j by a weighted average of all its neighbors in C_i,

$$e_j \approx \left(\sum_{k \in C_i} a_{jk} e_k \right) \Bigg/ \left(\sum_{k \in C_i} a_{jk} \right). \tag{2.4}$$

Substituting this into (2.3), we arrive at the AMG interpolation formula for the fine-grid point, i, as

$$e_i = - \sum_{k \in C_i} \left(\frac{a_{ik} + \sum_{j \in F_i^s} \frac{a_{ij} a_{jk}}{\sum_{l \in C_i} a_{jl}}}{a_{ii} + \sum_{j \in F_i^w} a_{ij}} \right) e_k. \tag{2.5}$$

With this definition of interpolation, the goals of the AMG coarse-grid selection process are clear. Each fine-grid point, i, should be strongly connected to (at least) one coarse-grid point, k, in order to take advantage of the property that $e_i \approx e_k$. Further, the requirement that each strongly connected fine-grid neighbor of i be itself strongly connected to some strongly connected coarse-grid neighbor of i must also be enforced. Finally, as with all multigrid schemes, there is the desire to make the coarse grid as small as possible, such that a good correction to the troublesome error components on the fine grid is still available. An initial coarse grid is selected as a maximal independent subset of the graph of strong connections [23]; thus, each fine-grid point must be strongly connected to at least one coarse-grid point, but the coarse set does not contain any pairs of strongly connected nodes. Then, a second pass of coarsening is performed, adding points to the tentative coarse grid from the first pass, ensuring that the necessary strong connections exist.

Finally, now that we have specified how to choose a coarse grid and interpolation from it, it remains to be seen how to restrict residuals to that grid and how to define an operator on the coarse grid. Both questions are answered by making use of the fact that the symmetric and positive-definite matrix, A, defines an inner-product and norm. Defining the A-inner product as $\langle \mathbf{u}, \mathbf{v} \rangle_A = \mathbf{v}^T A \mathbf{u}$ and the A-norm as $\|\mathbf{u}\|_A^2 = \mathbf{u}^T A \mathbf{u}$, the coarse-grid correction, $P\mathbf{e}_c$, that minimizes the A-norm of the corrected error satisfies $P^T A P \mathbf{e}_c = P^T(\mathbf{b} - A\mathbf{x})$. Thus, consistent with this variational principle, restriction is taken to be P^T, where P is the AMG interpolation operator, and the coarse-grid operator is chosen to be $P^T A P$.

3. AMG for complex-valued matrices.

Here, we consider the needed generalizations of the AMG components in the extension to the complex-valued case. In making these choices, we would like to design an algorithm that is consistent with the algorithm from section 2 for the case of a real-valued symmetric operator and that makes sense for the special cases of a complex-valued symmetric or Hermitian operator. In this section, we analyze these choices from an operator point of view.

3.1. Relaxation.

In generalizing the AMG algorithm to complex-valued matrices, we must ensure that relaxation performs as expected, in particular that (weighted) Jacobi and Gauss–Seidel relaxation are convergent for a reasonable class of problems and that they act as appropriate smoothers. Smoothing properties are discussed in detail in section 4. Just as classical AMG was originally proposed for M-matrices, for which the convergence of Jacobi and Gauss–Seidel is well understood [31, section 4.5], we consider here H-matrices, the complex generalization of M-matrices.

Definition 3.1. *Let $A \in \mathbb{C}^{n\times n}$ be such that its comparison matrix,*

$$(\mathcal{M}(A))_{ij} = \begin{cases} |a_{ii}| & \text{if } i = j, \\ -|a_{ij}| & \text{if } i \neq j, \end{cases}$$

is an M-matrix. Then, A is called an H-matrix.

For this class of matrices, the convergence of both weighted Jacobi and Gauss–Seidel relaxation is given in [29].

Theorem 3.2 (Theorem 1 from [29]). *For any nonsingular H-matrix, $A \in \mathbb{C}^{n\times n}$, let D be the diagonal of A and $-L$ be the strictly lower triangular part of A (so that $A - (D - L) = U$ is strictly upper triangular). Taking $J_\omega(A) = I - \omega D^{-1}A$ to be the error propagation operator for the weighted Jacobi iteration with weight ω and $G_\omega(A) = I - \omega(D - \omega L)^{-1}A$ to be the error propagation operator for the weighted Gauss–Seidel (SOR) iteration with weight ω, then*

- $\rho(J_1(A)) \leq \rho(J_1(\mathcal{M}(A))) < 1$,
- *for any* $\omega \in (0, \frac{2}{1+\rho(J_1(A))})$, $\rho(J_\omega(A)) \leq \omega\rho(J_1(A)) + |1-\omega| < 1$, *and*
- *for any* $\omega \in (0, \frac{2}{1+\rho(J_1(\mathcal{M}(A)))})$, $\rho(G_\omega(A)) \leq \omega\rho(J_1(A)) + |1-\omega| < 1$,

where $\rho(B)$ denotes the spectral radius of matrix B.

Note, in particular, that the first point of the theorem, convergence of the unweighted Jacobi iteration for both A and $\mathcal{M}(A)$, guarantees convergence of the under-relaxed weighted Jacobi iteration ($\omega \in (0,1)$) as stated in the second point. Similarly, the convergence of the (unweighted) Gauss–Seidel iteration is also guaranteed.

Obviously, the class of H-matrices is not the only class of complex-valued matrices for which Jacobi and Gauss–Seidel are convergent. However, as we are primarily interested in the performance of these schemes as smoothers, we would like to know more about the spectra of the Jacobi and Gauss–Seidel iteration matrices than simple bounds like those in Theorem 3.2 can give. As these spectra depend strongly on that of A, we will use LFA to gain more insight into smoothing in section 4.

3.2. Coarse-grid correction.

The definition of a good AMG coarse-grid correction scheme depends, of course, on the properties of the relaxation that it complements. While these properties are highly problem dependent, there are still certain broad principles that can guide AMG development. Central among these is that errors that are slow to be reduced by relaxation must lie in (or near to) the range of interpolation and that their residuals must be accurately restricted to the coarse grid. Here, we consider the components of the coarse-grid correction process independently and the principles that guide their selection.

Interpolation. Within the multigrid coarse-grid correction process, fine-grid errors are updated by the calculation

$$\mathbf{e}^{\text{new}} = (I - PB_c^{-1}RA)\mathbf{e}^{\text{old}},$$

where B_c^{-1} represents the approximate (or exact) inversion of the true coarse-grid operator, A_c. Such a correction affects only the parts of $\mathbf{e}^{\text{old}}$ that are in the range of P. Thus, the first principle for AMG coarse-grid correction does not change from the real-valued case; algebraically smooth errors must be in the range of P.

To accomplish this, standard AMG principles should be applied. For Hermitian and positive-definite matrices, the small-residual assumption (residuals of errors that are slow to be reduced by relaxation are small) again holds for Jacobi and Gauss–Seidel (see, for instance, [23, section 4.4]). Similarly, for Hermitian and positive-definite H-matrices, errors that are slow to be reduced by these relaxation schemes must vary slowly over connections where a_{ij} is large within row i of A. Thus, with a similar definition of strong connections, we can define interpolation for complex-valued matrices using (2.5), just as in the real-valued case. As for real-valued matrices, these requirements amount to assumptions on the class of matrices to which the complex AMG algorithm will be applied. If these assumptions are violated by the given problem, alternate techniques (such as the adaptive AMG algorithm [10]) should be used to define interpolation; see section 5.3.

Here, we use a simple extension of the classical AMG strong-connection measure, $S_i = \{j : |a_{ij}| \geq \theta \max_{k\neq i} |a_{ik}|\}$. This choice is justifiable for H-matrices, A, such that $\sum_j a_{ij} \approx 0$ for each i, similarly to the real case, where it is justifiable for M-matrices that satisfy the same conditions [23]. Under this assumption, it must also be the case that algebraically smooth errors vary slowly between strongly connected points. Once this definition is made, AMG coarse grids may be selected using a maximal independent set algorithm, as in classical AMG [23]. Choice of strong connections, and AMG coarsening in general, is still an area of active research [5, 8, 18].

It is interesting to note the relationship between multigrid approaches for nonsymmetric real matrices and the equivalent real form of a complex matrix. Writing $A \in \mathbb{C}^{n\times n}$ as $A = A^{(R)} + \imath A^{(I)}$ for $A^{(R)}$, $A^{(I)} \in \mathbb{R}^{n\times n}$, the complex system, $A\mathbf{u} = \mathbf{b}$, can be expressed in terms of its real parts as

$$\begin{bmatrix} A^{(R)} & -A^{(I)} \\ A^{(I)} & A^{(R)} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(R)} \\ \mathbf{u}^{(I)} \end{bmatrix} = \begin{bmatrix} \mathbf{b}^{(R)} \\ \mathbf{b}^{(I)} \end{bmatrix}, \tag{3.1}$$

where $\mathbf{u} = \mathbf{u}^{(R)} + \imath\mathbf{u}^{(I)}$ and $\mathbf{b} = \mathbf{b}^{(R)} + \imath\mathbf{b}^{(I)}$.

Dendy [14] suggests that for (nonsymmetric) matrices, interpolation should be built based on the symmetric part of the operator. This is motivated by considering convection–diffusion problems, where numerical experiments show that bilinear interpolation works well (when the second-order term is the constant-coefficient Laplacian), even when the convective term dominates. More recently, LFA has been used to confirm that this choice of interpolation works well for these problems [30]. For a Hermitian operator, the equivalent real form is symmetric (as $A_{ij}^{(I)} = -A_{ji}^{(I)}$) and, so, applying this principle results in no loss of generality. For a complex symmetric operator, on the other hand, the symmetric part of the equivalent real form is a block-diagonal matrix, and this principle suggests determining information based only on the real-part of A. Indeed, this approach has been investigated for complex matrices several times; cf. [16, 21].

We could consider generalizing this choice here, for example, by using the Hermitian part of the operator (which is the real part of a complex-symmetric matrix); however, such a choice would prejudice our algorithm toward the case where the differentially dominant operator occurs in the real part. In particular, consider a matrix arising in the discretization of $-\Delta u + \imath k^2 u = f$. Multiplying this equation by the complex unity, $\imath$, gives $-k^2 u - \imath\Delta u = \imath f$. Since the discrete Laplacian is symmetric (and not Hermitian), the Hermitian part of this operator corresponds only to the mass matrix, and any information about the diffusion term would be lost in interpolation. In principle, it seems wise to base interpolation on the differentially dominant term, if this is easily identified and extracted from the rest of the operator, but such a choice would not be consistent with the algebraic setting of the multigrid interpolation operator. In sections 4 and 5, we investigate these choices more thoroughly; however, because of examples such as that above, it appears that using only part of the matrix to generate interpolation is too restrictive to be applicable in all interesting cases as a black-box solver. Therefore, in section 5, we use the natural complex extension of (2.5) to define interpolation.

Restriction. Choosing restriction operators for the non-Hermitian definite case is more complicated, as no variational principle may be applied. One consideration for this choice is that the result of applying the rule to Hermitian-definite operators reduces to a variational approach when appropriate. Here, we propose several techniques for choosing restriction operators, motivated primarily by AMG considerations.

A common assumption in algebraic multigrid is that the residual vector after relaxation is small (close to zero), particularly at gridpoints associated only with the fine grid (the so-called F-points). This arises from a reduction-based multigrid (MGR) viewpoint [17, 22]. In MGR (or AMGr), relaxation is assumed to have an error propagation operator of the form $I - \begin{bmatrix} A_{ff}^{-1} & 0 \\ 0 & 0 \end{bmatrix} A$. After such a relaxation, the residual is exactly zero at the F-points. As the role of restriction is to transfer the residual from the fine grid to the coarse grid, this analysis suggests the choice of simple injection for restriction. Making this choice, however, is based on a rather extreme assumption that residuals at F-points are so small that they can be neglected entirely in the coarse-grid problem. The choice of restriction as injection is rarely used in AMG, particularly in the cases of Hermitian-definite or complex-symmetric operators, where the use of injection in the Galerkin product often leads to poor convergence.

Dendy suggests that restriction should be determined as the adjoint of interpolation for the adjoint of A [14], based on experiments with convection–diffusion problems. However, this idea may also be justified by considering a two-level (nonsymmetric) multigrid iteration with error-propagation operator,

$$T = (I - M_2^{-1}A)(I - PB_c^{-1}RA)(I - M_1^{-1}A), \tag{3.2}$$

where M_1 and M_2 represent the approximate inverses used in the (stationary) pre and postrelaxation steps and B_c represents the action of the coarse-grid solve process for some coarse-grid matrix, A_c. For a matrix, A, that is Hermitian definite, the usual variational conditions (that result in nonzero restriction weighting from the F-points) provide explicit guidance. In the general case, A itself cannot be used to define an appropriate norm, but the normal form, $A^\star A$, can (where $A^\star$ denotes the Hermitian transpose of A). Considering, then, the $A^\star A$ inner product and norm, we see that the adjoint of T in the $A^\star A$ inner product is $(A^\star A)^{-1}T^\star(A^\star A)$ and so

$$\|T\|_{A^\star A} = \|(A^\star A)^{-1}T^\star(A^\star A)\|_{A^\star A}.$$

Following (3.2), we can write

$$\begin{aligned}T^\star &= \left(I - A^\star (M_1^{-1})^\star\right)\left(I - A^\star R^\star (B_c^{-1})^\star P^\star\right)\left(I - A^\star (M_2^{-1})^\star\right)\\ &= A^\star\left(I - (M_1^{-1})^\star A^\star\right)\left(I - R^\star (B_c^{-1})^\star P^\star A^\star\right)\left(I - (M_2^{-1})^\star A^\star\right)A^{\star-1}\\ &= A^\star \overline{T}(A^\star)^{-1},\end{aligned}$$

where $\overline{T}$ takes the form of a two-grid cycle on $A^\star$, with the roles of R and P interchanged with their Hermitian transposes. Putting these together, we have that

$$\begin{aligned}\|T\|^2_{A^\star A} &= \max_{\mathbf{v}} \frac{\left\langle (A^\star A)^{-1} T^\star (A^\star A)\mathbf{v}, T^\star (A^\star A)\mathbf{v}\right\rangle}{\langle (A^\star A)\mathbf{v}, \mathbf{v}\rangle}\\ &= \max_{\mathbf{v}} \frac{\left\langle ((A^\star)^{-1} T^\star A^\star)A\mathbf{v}, ((A^\star)^{-1} T^\star A^\star)A\mathbf{v}\right\rangle}{\langle A\mathbf{v}, A\mathbf{v}\rangle}\\ &= \max_{\mathbf{w}} \frac{\langle \overline{T}\mathbf{w}, \overline{T}\mathbf{w}\rangle}{\langle \mathbf{w}, \mathbf{w}\rangle} = \|\overline{T}\|^2.\end{aligned}$$

Thus, the multigrid cycle given by T can be an effective cycle for A (measured in the $A^\star A$-norm) if and only if $\overline{T}$ is an effective cycle for $A^\star$ (measured in the ℓ_2-norm). But, to design an effective cycle for $A^\star$, we should apply the same principles to the choice of interpolation for this cycle (now $R^\star$) as we would for the cycle for A. In particular, the principle that $R^\star$ accurately represents the algebraically smooth errors of $A^\star$ should be enforced. In other words, $R^\star$ should be constructed as we would construct AMG interpolation for $A^\star$; R should be the adjoint of interpolation for the adjoint of A, just as was proposed for the real case in [14].

When A is also symmetric or Hermitian, this argument is consistent with typical multigrid approaches. If A is Hermitian, then $A^\star = A$, and this approach says that restriction should be the adjoint of AMG interpolation for A, $R = P^\star$. This is, of course, consistent with the variational conditions that typically guide multigrid development in the Hermitian-definite case. For complex-symmetric $A = A^{(R)} + \imath A^{(I)}$, $A^\star = A^{(R)} - \imath A^{(I)}$. If the rule for creating the AMG-style interpolation preserves this conjugation, then $P^\star(A^\star) = P^T(A)$. In other words, the choice of $R(A) = P^\star(A^\star)$ results in $R(A) = P^T(A)$ if A is complex symmetric and if

$$\begin{aligned}\Re\left(P(A^{(R)} + \imath A^{(I)})\right) &= \Re\left(P(A^{(R)} - \imath A^{(I)})\right)\\ \text{and } \Im\left(P(A^{(R)} + \imath A^{(I)})\right) &= -\Im\left(P(A^{(R)} - \imath A^{(I)})\right),\end{aligned} \tag{3.3}$$

where $\Re(M)$ denotes the matrix whose (i,j)th entry is the real part of m_{ij} and $\Im(M)$ is defined similarly for the imaginary part. In practice, this means that if the rule for determining interpolation only involves basic arithmetic operations (over which complex conjugation can be distributed) and the same points are selected as strong and weak connections for A and $A^\star$, then this rule results in a restriction operator that is the (non-Hermitian) transpose of interpolation.

A subspace decomposition point of view suggests a third approach for choosing restriction. When A is Hermitian and definite, a natural partition arises for $\mathbb{R}^n$, into the range of P and its A-orthogonal complement. In a two-level multigrid cycle, the coarse-grid correction stage exactly eliminates errors that lie in the range of P, while errors that are A-orthogonal to this space must be adequately reduced by relaxation on the fine grid. Let $\hat{P}$ be the ℓ_2-projection onto the range of the full-rank operator, P, and let A be Hermitian and definite. Then $\langle \hat{P}\mathbf{v}, \mathbf{w}\rangle_A = \langle \mathbf{v}, A^{-1}\hat{P}^\star A\mathbf{w}\rangle_A$. Thus, by the fundamental theorem of linear algebra, the space $\mathbb{R}^n$ may be partitioned as $\mathbb{R}^n = \mathcal{R}(\hat{P}) \oplus_A \mathcal{N}(A^{-1}\hat{P}^\star A)$. But, since $\hat{P}$ is a projection onto the range of P, $\mathcal{R}(\hat{P}) = \mathcal{R}(P)$. Furthermore, because $\hat{P}$ is an ℓ_2-projection, it is Hermitian, and since P has full rank, $\mathcal{N}(A^{-1}\hat{P}^\star A) = \mathcal{N}(P^\star A)$. Thus, we have that $\mathbb{R}^n = \mathcal{R}(P) \oplus_A \mathcal{N}(P^\star A)$.

Analyzing the error within the multigrid iteration using this subspace decomposition, we can identify those errors within the range of P as being the algebraically smooth errors. Thus, errors that are quickly attenuated by relaxation must lie in the null-space of $P^\star A$. Within the multigrid error propagation operator (as in (3.2)), we can then identify the role of the residual projection (application of $P^\star A$ or, in the non-Hermitian case, RA) as being to filter out those errors that can be easily treated through relaxation alone. As in this non-Hermitian case, A no longer defines a proper inner product; we can only consider the ℓ_2-adjoint of RA, $A^\star R^\star$ to use this analysis. Requiring that $\mathcal{N}(RA)$ includes all errors that are effectively reduced by relaxation is then equivalent to requiring that $\mathcal{R}(A^\star R^\star)$ includes the algebraically smooth errors. Thus, if only algebraically smooth errors are to be in the range of $A^\star R^\star$, then the small-residual assumption implies that $\mathcal{R}(A(A^\star R^\star))$ is small on fine-grid points. But $\mathcal{R}(A(A^\star R^\star)) = \mathcal{R}((AA^\star)R^\star)$, suggesting that $R^\star$ must be accurate for algebraically smooth errors of the normal equations, $AA^\star$. This leads to another possible rule for defining restriction, as the Hermitian conjugate of an AMG interpolation for the normal operator, $AA^\star$.

Such a choice, while motivated by typical AMG considerations, is not as attractive from a cost perspective as those discussed previously. The costs of forming $AA^\star$ in order to form restriction are obviously significant and would almost certainly lead to an increase in complexity of the AMG coarse-grid operators if applied within (2.5). On the other hand, if the basic AMG interpolation scheme is adapted to such complications, then this approach can be quite effective. Investigation of a similar approach within smoothed aggregation multigrid is currently underway [11].

Forming the coarse-grid operator. In many cases, physical intuition may be used to define an appropriate coarse-grid operator that complements the given choices of interpolation and restriction, but this is difficult to use consistently in the algebraic setting considered here. Instead, we choose the obvious generalization of the Galerkin condition from the symmetric or Hermitian definite case and we define $A_c = RAP$. In particular, this can be viewed as a restriction of the fine-grid operator, A, to exactly those components identified as needing correction from the coarse grid (the algebraically smooth errors). Multiplication on the right serves to restrict the domain of A to the range of P that, by assumption, contains these errors. Multiplication on the left by R restricts the range of A to that of R, which may be chosen, as described above, based on an understanding of the action of A on algebraically smooth errors. Using the definition of R discussed above, this choice for the coarse-grid operator also preserves the Hermitian symmetry or regular symmetry (if conditions (3.3) are satisfied) of the fine-grid operator. While it is possible to make separate choices of restriction and interpolation for use in the Galerkin product and in the multigrid cycle, in this paper we choose the same R and P for both roles.

3.3. Relation to systems AMG.

While the proposed approach directly treats the complex values in the given matrix, it is also possible to implement this approach indirectly, within an existing AMG code that allows so-called point-based treatment of systems of equations. Viewing the equivalent real form (3.1) as a system of real-valued equations with two unknowns per discrete node point (the real and imaginary parts of the complex value), decisions within AMG can be made based on the 2 × 2 blocks that represent $a_{ij} = a_{ij}^{(R)} + \imath a_{ij}^{(I)}$ as

$$A_{ij} = \begin{bmatrix} a_{ij}^{(R)} & -a_{ij}^{(I)} \\ a_{ij}^{(I)} & a_{ij}^{(R)} \end{bmatrix}.$$

In this setting, pointwise relaxation on the complex form of A is equivalent to block relaxation, simultaneously solving the 2 × 2 blocks associated with each node point. The definition of strong connections is recovered using the L^2-norm of the block matrices, A_{ij}, in place of $|a_{ij}|$. Interpolation (and, thus, restriction) may be defined using block matrices (and block inverses), following (2.5) to define the 2 × 2 nodal blocks of interpolation. Finally, the Galerkin condition also can be computed in this block form.

The extra costs of such an approach, however, make the complex-valued approach proposed here attractive, especially for the very large systems that arise in many naturally complex-valued applications. A naive implementation of the block algorithm would require twice the storage and twice the work to compute a matrix-vector product as the complex-valued AMG algorithm does. Furthermore, increased work would also be necessary in the AMG setup algorithm, unless modifications to the systems AMG code were made to take advantage of the complex structure. With such modifications, however, the computations performed by the systems AMG algorithm would simply mimic those described here.

4. LFA.

Since the early days of multigrid, Fourier smoothing and two-grid analyses have been used to make quantitative estimates of the smoothing properties of basic iterative methods and for quantitative evaluation of the other multigrid components in a two-grid method; see, for example [3, 25, 26]. LFA (called local mode analysis in [3]) is the main multigrid analysis option for problems that do not lead to Hermitian and positive-definite matrices, such as the complex-valued problems of interest here. As we are interested in the definition of the coarse-grid correction components within a multigrid cycle, smoothing analysis alone is not sufficient, and we also consider two- and three-grid LFA [30]. Especially in this complex-valued setting, increased insight into the quality of the transfer-operator-dependent Galerkin coarsening is of value.

Here, three-grid analysis is briefly outlined for two-dimensional problems with standard coarsening. We consider a discrete problem, $A_h \mathbf{u}_h = \mathbf{f}_h$, where $\mathbf{u}_h$ represents the exact discrete solution on a regular grid with mesh size h. The main idea in the Fourier analysis is to formally extend all multigrid components to an infinite grid, $G_h := \{\boldsymbol{x} = (k_x h, k_y h) : k_x, k_y \in \mathbb{Z}\}$. On G_h, we have a unitary basis of grid functions called the Fourier components,

$$\varphi_h(\boldsymbol{\theta}, \boldsymbol{x}) := \exp(\imath \boldsymbol{\theta} \cdot \boldsymbol{x}/h) = \exp(\imath \boldsymbol{k} \cdot \boldsymbol{\theta})$$

with $\boldsymbol{x} \in G_h$, $\boldsymbol{k} = (k_x, k_y)$, and Fourier frequencies, $\boldsymbol{\theta} = (\theta_x, \theta_y) \in \mathbb{R}^2$. These components are eigenfunctions of any discrete, real- or complex-valued operator, A_h, on G_h with constant coefficients.

Recall that the error, $\mathbf{e}_h^m = \mathbf{u}_h^m - \mathbf{u}_h$, after iteration m is transformed by a two-grid operator as

$$\begin{gathered}\mathbf{e}_h^{m+1} = T_h^{2h}\,\mathbf{e}_h^m \quad \text{with} \\ T_h^{2h} = S_h^{\nu_2} K_h^{2h} S_h^{\nu_1} \quad \text{and} \quad K_h^{2h} = I_h - P_{2h}^h (A_{2h})^{-1} R_h^{2h} A_h,\end{gathered} \tag{4.1}$$

and, after a three-grid cycle, is given by

$$\begin{gathered}\mathbf{e}_h^{m+1} = T_h^{4h}\,\mathbf{e}_h^m \quad \text{with} \\ T_h^{4h} = S_h^{\nu_2} K_h^{4h} S_h^{\nu_1} \quad \text{and} \quad K_h^{4h} = I_h - P_{2h}^h (I_{2h} - (T_{2h}^{4h})^\gamma)(A_{2h})^{-1} R_h^{2h} A_h.\end{gathered} \tag{4.2}$$

Here, T_{2h}^{4h} defined by (4.1) reads $T_{2h}^{4h} = S_{2h}^{\nu_2}(I_{2h} - P_{4h}^{2h}(A_{4h})^{-1} R_{4h}^{2h}) S_{2h}^{\nu_1}$. A_h, A_{2h}, and A_{4h} correspond to discretizations on the h-, $2h$-, and $4h$-grids, although A_{2h} and A_{4h} may also be based on Galerkin principles, as described above. S_h and S_{2h} are the smoothing operators on the fine and the first coarse grid, and ν_i $(i = 1, 2)$ represents the number of pre- and postsmoothing steps. R_h^{2h}, R_{2h}^{4h} and P_{2h}^h, P_{4h}^{2h} denote restriction and prolongation operators, respectively, between the different grids. I_h and I_{2h} are the identity operators with respect to the h- and the $2h$-grids.

Instead of inverting A_{2h}, as is done in (4.1), the $2h$-equations are solved approximately in a three-grid cycle (4.2) by performing γ two-grid iterations, T_{2h}^{4h}, with zero initial approximation. This is reflected by the replacement of $(A_{2h})^{-1}$ from (4.1) by the expression $(I_{2h} - (T_{2h}^{4h})^\gamma)(A_{2h})^{-1}$.

In two-grid Fourier analysis, we distinguish between low and high frequencies,

$$\Theta_{low}^{2g} = (-\pi/2, \pi/2]^2 \quad \text{and} \quad \Theta_{high}^{2g} = (-\pi, \pi]^2 \setminus \Theta_{low}^{2g},$$

in such a way that the low-frequency components are "visible" on both grids G_h and G_{2h}. Each low-frequency component is coupled with three related high-frequency components that alias on G_{2h}, leading to a splitting of the Fourier space into four-dimensional subspaces, the spaces of $2h$-harmonics:

$$\begin{aligned}&\operatorname{span}\{\varphi(\boldsymbol{\theta}^{\boldsymbol{\alpha}}, \boldsymbol{x});\ \boldsymbol{\alpha} = (\alpha_x, \alpha_y),\ \alpha_x, \alpha_y \in \{0, 1\}\} \quad \text{with} \\ &\boldsymbol{\theta} = \boldsymbol{\theta}^{00} \in \Theta_{low}^{2g} \ \text{ and } \ \boldsymbol{\theta}^{\alpha_x \alpha_y} := (\theta_x - \alpha_x \operatorname{sign}(\theta_x)\pi, \theta_y - \alpha_y \operatorname{sign}(\theta_y)\pi).\end{aligned}$$

T_h^{2h} is unitarily equivalent to a block diagonal matrix consisting of 4×4 blocks. This simple representation is then used to calculate the corresponding spectral radius and, thus, the LFA two-grid convergence factor, ρ_{2g}. The smoothing factor, μ, which measures the reduction of high-frequency error components by relaxation is defined based on a coarse-grid correction operator in (4.1) that annihilates the low-frequency error components. K_h^{2h} is thus replaced in (4.1) by a projection, Q_h^{2h}, onto the space of high frequencies.

Similar to the two-grid case, in three-grid LFA we distinguish between low and high frequencies, but now with respect to three grids, G_h, G_{2h}, and G_{4h}. It is then appropriate to divide the Fourier space into a direct sum of 16-dimensional subspaces, the so-called $4h$-harmonics [30]. As a consequence, T_h^{4h} is unitarily equivalent to a block diagonal matrix with at most 16×16 blocks. We obtain the LFA three-grid convergence factor, ρ_{3g}, as the supremum of the spectral radii of the block matrices. The assumptions needed for LFA to be valid seem far from the algebraic setting considered here; however, the construction of the complex-valued smoothing and coarse-grid correction components can still be guided by the LFA results. In particular, the algorithm should also converge well for structured-grid problems, under the usual LFA restrictions. In this sense, LFA results serve as a first indication of the quality of the coarse-grid correction proposed.

4.1. LFA results. As Fourier analysis applies only to structured grids, we analyze a structured-grid variant of the AMG interpolation described above. The natural structured-grid variant of AMG interpolation is that used in Dendy's black-box multigrid (BMG); for details, see [2, 14]. For restriction, our operator of choice is

$$R_h^{2h} = [P_{2h}^h]^\star (A^\star), \tag{4.3}$$

as in section 3.2. Next to this choice of restriction, we also include the straight injection operator in our evaluation. The third option discussed in section 3.2 with restriction based on $AA^\star$ is not considered here due to the expense needed to calculate it.

We start the LFA experiments with a Laplacian-type operator,

$$A \overset{\wedge}{=} \begin{bmatrix} & -1 & \\ \imath & 4 & \imath \\ & -1 & \end{bmatrix}.$$

We fix two red–black Jacobi relaxation sweeps as the smoother with $\omega = 0.9$ (as discussed below) and compare the performance of the simple real-valued transfer operators of full-weighting restriction (FW) and bilinear interpolation (BL) to the complex-valued BMG interpolation and restriction, based on (4.3) and on injection (INJ). Table 4.1 gives the LFA two- and three-grid convergence factors. The LFA smoothing factor for this red–black relaxation is $\mu^2 = 0.217$.

We see in Table 4.1 that the complex-valued transfer operators perform satisfactorily, slightly better than the real-valued transfer operators. Injection also performs well on this problem, giving superior results with BMG interpolation, even in combination with red–black relaxation and five point stencils. Similar behavior is seen when the red–black Jacobi relaxation is replaced by a forward–backward Gauss–Seidel relaxation.

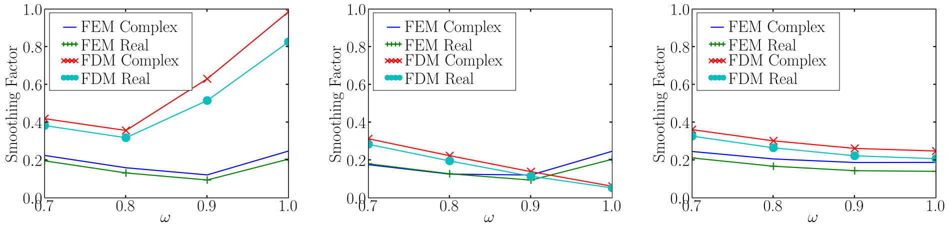
We next consider a definite Helmholtz operator, $-\Delta u + \alpha u$, discretized either by standard finite differences or by bilinear finite elements on a uniform mesh, leading to the standard $O(h^2)$ discretization stencils.

Figure 4.1 displays smoothing factors for this problem, with $\alpha = k^2$ (real) and $\alpha = k^2\imath$ (complex), and their dependence on the relaxation parameter ω (commonly used in multigrid smoothers). Three smoothers are compared: pointwise damped Jacobi, damped red–black Jacobi (which is identical to red–black Gauss–Seidel for five-point stencils) and lexicographical Gauss–Seidel (a forward sweep followed by a backward sweep) with $\nu = 2$ smoothing steps. One forward–backward pair of Gauss–Seidel sweeps is considered as two smoothing steps. We compare in each

Table 4.1

LFA two- and three-grid convergence factors for a Laplacian with complex entries.

	BL-FW	BL-INJ	BMG-BMG	BMG-INJ
ρ_{2g}	0.217	0.299	0.188	0.158
ρ_{3g}^V	0.217	0.310	0.188	0.158



(a) Pointwise Jacobi

(b) Red–black Jacobi

(c) Forward–backward Gauss–Seidel

FIG. 4.1. *Dependence of LFA smoothing factors ($\nu = 2$) on the relaxation parameter, ω, for Jacobi, red–black Jacobi, and forward–backward Gauss–Seidel smoothers.*

TABLE 4.2
LFA smoothing, two- and three-grid factors for the FE discrete complex Helmholtz operator.

	ω-JAC $\omega = 0.9$	ω-JAC-RB $\omega = 0.9$	GS-FWBW $\omega = 1.0$
μ^2	0.12	0.12	0.18
ρ_{2g}	0.14	0.15	0.17
ρ_{3g}^V	0.35	0.29	0.17

subfigure the finite difference and finite element stencils for problems with positive real- or complex-valued Helmholtz terms, $\alpha = k^2$ or $\alpha = k^2\imath$. Parameters are set as $h = 1/64, k^2 = 1600$.

Table 4.2 presents two- and three-grid LFA convergence factors for V(1,1) cycles applied to the FE discretization of the complex-valued Helmholtz operator with complex-valued transfer operators. The LFA three-grid V-cycle factors show degradation for standard and red–black Jacobi relaxation, which is an indication that the coarse-grid problems are not defined optimally for these smoothers. A closer look at the Galerkin coarse-grid operators built with these transfer operators shows that on the third grid, operators with only positive elements arise. While the convergence with Jacobi relaxation degrades, Gauss–Seidel relaxation is not influenced by these coarse-grid discretizations and performs satisfactorily.

Finally, we mention that the use of injection as the restriction operator for this Helmholtz operator did not lead to satisfactory LFA factors. The two- and three-grid factors increase to at least 0.56 for different smoothers and interpolation operators, indicating the advantages of nontrivial restriction.

Remark 1. A remark on the need for complex-valued interpolation follows.

The LFA assumptions of full coarsening and constant stencils are more restrictive than our expectations for AMG. While LFA does provide useful insight into several properties of multigrid for complex systems, our analysis cannot distinguish between the benefits of real-valued and complex-valued interpolation operators. Instead, we provide a simple example (a special case of the operators considered in section 5.3) to demonstrate the benefits of a "fully complex" AMG approach.

Consider the Hermitian matrix, A, defined over a two-dimensional mesh by

$$4u_{i,j} - e^{-i\phi_{i-1,j}}u_{i-1,j} - e^{i\phi_{i,j}}u_{i+1,j} - e^{-i\psi_{i,j-1}}u_{i,j-1} - e^{i\psi_{i,j}}u_{i,j+1} = f_{i,j},$$

where the fields, $\{\phi_{i,j}\}$ and $\{\psi_{i,j}\}$, are chosen randomly. Applying the AMG strength-of-connection test to this stencil suggests that all neighboring points are strongly connected, as all off-diagonal entries are of the same size. As a result, AMG naturally chooses a structured red–black coarsening pattern. In this setting, two-level AMG with complex interpolation operators is an exact solver; partitioning $A = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{fc}^{\star} & A_{cc} \end{bmatrix}$, A_{ff} is diagonal, and AMG naturally chooses the complex interpolation operator,

$$P = \begin{bmatrix} -A_{ff}^{-1}A_{fc} \\ I \end{bmatrix}.$$

By contrast, choosing a real-valued interpolation operator, say,

$$P = \begin{bmatrix} A_{ff}^{-1}|A_{fc}| \\ I \end{bmatrix},$$

where $|M|$ denotes the matrix with entries $|m_{ij}|$, leads to a two-level convergence factor of 0.585. This discrepancy remains in three-level and multilevel convergence factors (0.155 with complex interpolation, 0.589 with real interpolation). LFA based on red–black coarsening may give further insight into this choice and is a question for future research. □

5. Numerical results. All numerical experiments are run on a 64-bit AMD Athlon 3700+ system, running at 2.2 GHz, with 3 GB of RAM. We use the standard gnu compiler collection (gcc) C-compiler with appropriate optimization options enabled for these machines. This compiler supports the C99 complex standard and, so, we use the native complex arithmetic functions to implement the algorithms described above. Systems on the coarsest level of the multigrid hierarchy are solved using a direct solver (LAPACK's `zgbtrs` routine); in all examples, multigrid coarsening is continued until the coarsest level is so small that this is a negligible part of the iteration cost. For all examples, we consider V(1,1) multigrid cycles, using Gauss–Seidel relaxation ordered so that the coarse-grid points are relaxed first on the downward sweep of the V-cycle and last on the upward sweep (the so-called $CF-FC$ ordering) with (complex-valued) interpolation chosen as the generalization of (2.5) from the real-valued case.

5.1. Simple problems. First, we consider variants of several simple problems for which standard multigrid and AMG performance are well understood, in order to demonstrate that the generalization to complex arithmetic maintains these properties. Additionally, this provides a benchmark for comparison of the costs of AMG in real arithmetic versus complex arithmetic.

Table 5.1 shows the performance of real-valued AMG for bilinear finite element discretizations of the positive-definite Poisson equation, with and without a positive-definite shift, $-\Delta u = f$ and $-\Delta u + k^2 u = f$, with $k = 0.625/h$. The coarse-grid

TABLE 5.1
Real-valued AMG performance for finite-element Poisson, with and without a definite shift.

		$-\Delta u$		$-\Delta u + k^2 u$	
		standard S_i	modulus S_i	standard S_i	modulus S_i
512×512	ρ_{MG}	0.116	0.116	0.041	0.041
	# Iters.	7	7	6	6
	t_{solve}	2.6	2.6	2.6	2.4
1024×1024	ρ_{MG}	0.136	0.136	0.041	0.041
	# Iters.	7	7	6	6
	t_{solve}	10.4	10.4	10.4	9.8

Table 5.2
Complex-valued AMG performance for simple problems.

		$-\Delta u$	$-\imath\Delta u$	$-\Delta u + k^2 u$	$-\Delta u + \imath k^2 u$
512×512	ρ_{MG}	0.116	0.116	0.041	0.171
	# Iters.	7	7	6	11
	t_{solve}	4.0	4.0	3.8	5.3
1024×1024	ρ_{MG}	0.136	0.136	0.041	0.172
	# Iters.	7	7	6	12
	t_{solve}	17.5	16.6	15.4	22.5

selection is based on a strong-connection definition of either $-a_{ij} \geq \theta \max_{k \neq i}\{-a_{ik}\}$ (standard) or $|a_{ij}| \geq \theta \max_{k \neq i}\{|a_{ik}|\}$ (modulus-based) for $\theta = 0.25$ and, here, we take $R = P^T$ and $A_c = P^T AP$. Shown in Table 5.1 are the maximum convergence factor observed over (up to) 200 iterations, as well as the iteration count and total time needed to reduce the ℓ_2 norm of the residual by a relative factor of 10^9 for each problem.

For the unshifted Poisson problem, there is no difference in the results using the standard or modulus-based definition of strong connections, due to the M-matrix structure of the finite-element operators on these regular meshes. This is preserved on coarse meshes, so that the standard and modulus-based definitions coincide, and the performance of the two approaches is identical. The same is not true for the shifted problem, where the coarsening of the mass matrix induces positive off-diagonal entries in the coarse-grid operators. As a result, the cost of a multigrid V(1,1) cycle is slightly lower for the modulus-based measure of strength of connection, leading to slightly faster times for that approach.

Table 5.2 details the performance of the complex-valued AMG algorithm for these problems, along with two simple complex generalizations. For the real-valued problems, we see that the complex AMG solver performs the same as the real-valued AMG solver does (cf. Table 5.1) when measured in terms of convergence factors, ρ_{MG} or iteration counts. In terms of CPU time, however, we see that there is a premium to be paid for doing complex arithmetic; however, the cost is only 50% to 70% greater than that of the real-valued AMG algorithm. For the complex Poisson operator, $-\imath\Delta u$, performance of complex AMG essentially matches that of both the real and complex AMG algorithms applied to the usual Poisson operator. Different results are seen for the complex-shifted Helmholtz operator, $-\Delta u + \imath k^2 u$, where the convergence factors (while still bounded nicely away from 1.0) increase somewhat. Comparing to Figure 4.1, we see that the solver performance for the complex-shifted problem is quite close to that predicted by LFA with lexicographic relaxation order, while the performance for the real shift is much better. Using lexicographic-ordered Gauss–Seidel relaxation for the positive shift leads to performance similar to that predicted by LFA. For this problem, however, an unsymmetric ordering of relaxation offers a significant improvement over lexicographic ordering.

The complexity of the algebraic multigrid iterations may be measured in terms of the AMG grid complexity, c_g, and operator complexity, c_A. The grid complexity, defined as the sum of the number of grid points on all levels in the AMG hierarchy divided by the number of grid points on the first level, gives a measure of the storage costs needed for coarse-level approximations, residuals, and right-hand sides during the multigrid iteration. The operator complexity, defined as the sum of the number of nonzeros in the system matrices defined on all levels of the AMG hierarchy divided by the number of nonzeros in the fine-grid operator, gives both a measure of the storage needed for the coarse-grid operators within the multigrid hierarchy and of the cost per multigrid V-cycle, relative to that of a fine-grid matrix-vector multiply. For real-valued AMG, we measure these complexities in terms of real values stored, while for complex-valued AMG, we measure these complexities by the number of complex values stored. For the problems considered in this section, all grid and operator complexities are quite low. Using real-valued AMG with the standard definition of S_i for the shifted problem resulted in the largest complexities, $c_g = 1.38$ and $c_A = 1.63$. For all other problems considered here, grid and operator complexities were nearly uniform, with $c_g = 1.33$ and $c_A = 1.41$. These are typical of geometric multigrid complexities for the regular stencil patterns and structured grids considered in this section.

5.2. Unstructured grid application. In this section, we consider the effect of unstructured grids on the performance of the complex-valued AMG algorithm. The discrete problems arise from a linear finite-element discretization of a Helmholtz problem with complex shift that arises from a reduction of Maxwell's equations [16]. As a result, the discrete problem is complex symmetric; thus, we consider complex AMG with the choice of $R = P^T$ as discussed above, along with preconditioning of BiCGStab (as CG is no longer a suitable choice).

In the special case of a time-harmonic source current, Maxwell's equations may be reduced to a frequency-domain Helmholtz equation for the z-component of the Fourier transform of a vector potential, $\mathbf{A}$. Details of this reduction can be found, for example, in [16], resulting in the equation for $\hat{A}_z$,

$$-\nabla \cdot \left(\frac{1}{\mu} \nabla \hat{A}_z \right) + \imath\omega\sigma \hat{A}_z = \hat{J}_{s,z}.$$

We consider only half an annular domain, as depicted in Figure 5.1, discretized using standard linear finite elements. Lahaye et al. solve these systems using a real-valued AMG algorithm (based on the real part of the system matrix) as a preconditioner for BiCGStab [16].

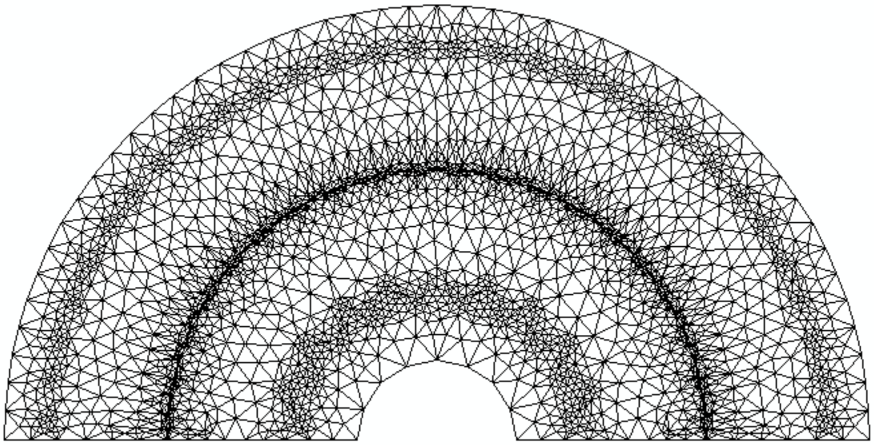


FIG. 5.1. *Mesh geometry for induction motor. Reprinted with permission from D. Lahaye, H. DeGersem, S. Vandewalle, and K. Hameyer, Algebraic multigrid for complex symmetric systems, IEEE Trans. Magn.,* 36 (2000), *pp.* 1535–1538. *Copyright* ©2000 *IEEE.*

Table 5.3
AMG performance for finite-element models of induction motor.

Problem	Solver	c_g	c_A	t_{setup}	t_{solve}	# Iters.
1028 nodes $nnz = 6520$	real AMG	1.80	2.17	0.0	0.0	19
	complex AMG	1.80	2.17	0.0	0.0	20
	AMG-BiCGStab	1.80	2.17	0.0	0.0	6.5
	cAMG-BiCGStab	1.80	2.17	0.0	0.0	6
3959 nodes $nnz = 26601$	real AMG	1.85	2.67	0.0	0.1	29
	complex AMG	1.85	2.67	0.0	0.1	39
	AMG-BiCGStab	1.85	2.67	0.0	0.1	8.5
	cAMG-BiCGStab	1.85	2.67	0.0	0.1	9
15302 nodes $nnz = 104926$	real AMG	1.83	2.86	0.1	0.6	29
	complex AMG	1.82	2.85	0.2	0.7	32
	AMG-BiCGStab	1.83	2.86	0.1	0.4	9
	cAMG-BiCGStab	1.82	2.85	0.2	0.3	8
34555 nodes $nnz = 239661$	real AMG	1.81	2.91	0.4	1.7	31
	complex AMG	1.81	2.91	0.4	1.7	30
	AMG-BiCGStab	1.81	2.91	0.4	1.0	8.5
	cAMG-BiCGStab	1.81	2.91	0.4	1.0	8.5
75951 nodes $nnz = 529317$	real AMG	1.77	2.87	1.0	4.5	31
	complex AMG	1.77	2.87	1.1	4.2	29
	AMG-BiCGStab	1.77	2.87	1.0	2.6	8.5
	cAMG-BiCGStab	1.77	2.87	1.1	2.5	8

We consider five different resolutions on the half-annulus geometry of Figure 5.1. The triangulation on the coarsest mesh has 1028 nodes, while subsequent meshes are refinements of this initial triangulation. The finest mesh has 75,951 nodes and approximately 530,000 nonzero entries in the system matrix. For each mesh, we consider the performance of ILU-preconditioned BiCGStab, AMG based on the real-part of the matrix and based on the complete, complex matrix, both as a standalone solver and as a preconditioner for BiCGStab.

The performance of the AMG variants for these problems is detailed in Table 5.3. For each problem and each AMG approach, we measure both the grid complexity, c_g, and operator complexity, c_A, of the complex AMG solver. Additionally, we report setup and solve times (in seconds), as well as the number of iterations needed to reduce the ℓ_2-norm of the residual by a relative factor of 10^9. Because of the small sizes of the least-refined meshes, some times are below the threshold that can be accurately measured; such times are reported as 0.0.

Table 5.3 shows that the complex-valued AMG achieves performance similar to that seen with the real-valued AMG preconditioning investigated in [16]. In particular, the number of iterations required to reduce the residual from a zero initial guess by a relative factor of 10^9 are quite close to those of an AMG algorithm based solely on the (real) differential part of the operator. In timing these results, we have not optimized the distribution of real- and complex-valued arithmetic in the real-valued AMG case. This means that, in practice, preconditioning based solely on the real part of the operator is more efficient, given the added efficiency possible using real-valued storage for the interpolation operators.

For comparison, we consider the performance of BiCGStab preconditioned with ILU(0) for these problems. Note that because of the fixed nonzero structure in the preconditioner (matching that of A), the effective operator complexity for these preconditioners is 2, as both the original matrix and its ILU factors must be computed and stored. Because of the simple calculation of this factorization, setup times for these preconditioners are negligible. Iteration costs, however, are significant, with over 6000 iterations needed for the second-largest grid and stalling convergence on the largest problem.

5.3. Gauge Laplacian. While examples such as the problems considered in the previous subsections are interesting, they do not require the full strength of the complex AMG algorithm introduced here. In order for a complex-valued approach to be a necessity, the dynamics of the problem must be somehow inherently complex valued, whereas for problems such as the complex-shifted Helmholtz equation, the dominant part of the operator is, in the end, real-valued. In this section, we discuss model problems from a large class of applications with inherently complex dynamics, related to the numerical simulation of quantum field theory [20, 24].

Quantum chromodynamics (QCD) is the part of the standard model of physics that describes the strong interaction between quarks and gluons within particles such as protons and neutrons. In order to test the validity of the standard model, QCD is used to make predictions of properties of known particles, which may be independently measured using a particle accelerator [12]. Discretizing QCD requires the computation of a Feynman path integral over certain Grassmann (anticommuting) variables, which may be simplified by introducing an effective gauge field [20]. Such a simplification, however, introduces an inverse of the Dirac operator. A full description of this problem is beyond the scope of this work; we refer the interested reader to [20, 24] and will concentrate here on a model problem that displays many of the same complications as the discretization of the full Dirac operator.

The Dirac equation is a first-order system of 12 coupled PDEs posed on a four-dimensional space. The 12 variables (scalar-functional degrees of freedom), however, appear as a tensor-product of a four-dimensional space (associated with the quantum dynamical spin) with a three-dimensional space (associated with the quantum dynamical color). The Dirac operator is thus block-structured, with differential coupling between variables of the same spin index but only algebraic coupling between variables of different spins. Exploiting this structure, the Dirac operator may be written as

$$M(A) = \sum_{\mu=1}^{4} (\gamma_\mu \otimes (I_3 \partial_\mu - \imath A_\mu)) - mI_{12},$$

where $\mu = 1, \ldots, 4$ denote the four canonical space–time directions, $\{\gamma_\mu\}_{\mu=1}^{4}$ are four fixed unitary matrices with all entries 0,± 1, and $\pm\imath$, ∂_μ is the standard partial derivative in the μ-direction, I_3 and I_{12} are the 3×3 and 12×12 identity matrices, respectively, the constant, m, is a mass parameter, and $\otimes$ denotes a standard tensor product of operators. The field of complex 3×3 matrices, $A_\mu(\mathbf{x})$, is known as the gauge potential and is chosen through a Monte Carlo process in the numerical simulation of QCD.

Here, we consider a two-dimensional model related to the Dirac equation, with a scalar potential, $a(\mathbf{x}, \mu)$ (where $\mathbf{x}$ now varies over a two-dimensional lattice, and $\mu = 1, 2$). Considering the covariant derivative term, $D_\mu = \partial_\mu - \imath a(\mathbf{x}, \mu)$, we can discretize this term over the lattice by integrating its action along the edge between two lattice points. Just as $\int_{\mathbf{x}_i}^{\mathbf{x}_j} \partial_\mu \psi(\mathbf{x}) d\mathbf{x}_\mu = \psi(\mathbf{x}_j) - \psi(\mathbf{x}_i)$ leads to a standard finite-difference discretization of the first derivative, ∂_μ, we may approximate

$$\int_{\mathbf{x}_i}^{\mathbf{x}_j} \left(\partial_\mu - \imath a(\mathbf{x},\mu)\right)\psi(\mathbf{x})d\mathbf{x}_\mu = \int_{\mathbf{x}_i}^{\mathbf{x}_j} e^{\imath a(\mathbf{x},\mu)}\partial_\mu\left(e^{-\imath a(\mathbf{x},\mu)}\psi(\mathbf{x})\right)d\mathbf{x}_\mu$$
$$\approx e^{\imath a(\hat{\mathbf{x}},\mu)}\left(e^{-\imath a(\mathbf{x}_j,\mu)}\psi(\mathbf{x}_j) - e^{-\imath a(\mathbf{x}_i,\mu)}\psi(\mathbf{x}_i)\right).$$

Choosing $\hat{\mathbf{x}} = \mathbf{x}_i$ leads to the approximation

$$D_\mu\psi(\mathbf{x}) \approx \frac{1}{h}\left(e^{\imath(a(\mathbf{x}_i,\mu)-a(\mathbf{x}_j,\mu))}\psi(\mathbf{x}_j) - \psi(\mathbf{x}_i)\right),$$

while choosing $\hat{\mathbf{x}} = \mathbf{x}_j$ leads to

$$D_\mu\psi(\mathbf{x}) \approx \frac{1}{h}\left(\psi(\mathbf{x}_j) - e^{\imath(a(\mathbf{x}_j,\mu)-a(\mathbf{x}_i,\mu))}\psi(\mathbf{x}_i)\right).$$

Defining $\alpha(\mathbf{x}_i,\mathbf{x}_j) = a(\mathbf{x}_i,\mu) - a(\mathbf{x}_j,\mu)$ to be the weighting of the covariant derivative over the lattice edge between nodes i and j, we see that these two choices of $\hat{\mathbf{x}}$ lead to closely related forward and backward difference formulae. Defining the second derivative stencil as the weighted difference of these forward and backward differences (to define the first derivatives at $\mathbf{x}_{i\pm\frac{1}{2}}$, assuming nodes $\mathbf{x}_i$, $\mathbf{x}_{i\pm1}$ are adjacent), we have

$$-D_\mu^2\psi(\mathbf{x}_i) \approx \frac{1}{h^2}\left(-e^{-\imath\alpha(\mathbf{x}_{i-1},\mathbf{x}_i)}\psi(\mathbf{x}_{i-1}) + 2\psi(\mathbf{x}_i) - e^{\imath\alpha(\mathbf{x}_i,\mathbf{x}_{i+1})}\psi(\mathbf{x}_{i+1})\right).$$

Proper specification of the gauge potential is needed in order to appropriately model QCD applications. Here, we consider the case of a unit lattice spacing ($h = 1$) and take $\alpha(\mathbf{x}_i,\mathbf{x}_j)$ to be a random variable of the form $\alpha(\mathbf{x}_i,\mathbf{x}_j) = 2\pi\beta\theta(\mathbf{x}_i,\mu)$, where β is a temperature parameter and $\theta(\mathbf{x}_i,\mu)$ is chosen independently for each node, $\mathbf{x}_i$, and direction, μ, on the lattice from a normal distribution with mean 0 and variance 1. We consider doubly periodic two-dimensional lattices. For $\beta = 0$, this recovers the positive-semidefinite five-point finite-difference Laplacian; for $\beta > 0$, the matrices are positive definite and Hermitian.

Figure 5.2 shows the convergence factors for three variants of multigrid applied to the discrete covariant Laplacian on a 513×513 grid as β increases. The simplest

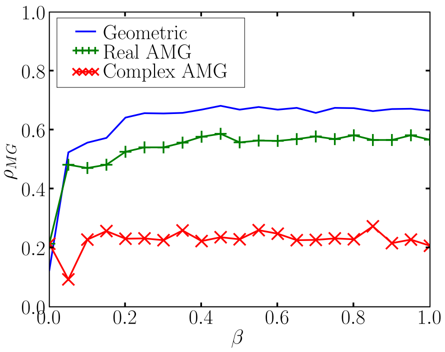


FIG. 5.2. *Convergence factors for covariant Laplacian with varying β.*

variant is multigrid based on full coarsening, with bilinear interpolation and full-weighting restriction. The method of [16, 21] is also considered, where the classical (Ruge–Stüben) AMG coarsening and interpolation are performed based on the real part of the matrix, denoted by "real AMG." Finally, the complex AMG proposed here is also tested. For $\beta = 0$, the matrix is the usual five-point periodic Laplacian, and all three methods perform well. As β increases, however, the performance of geometric multigrid and real AMG quickly degrade, confirming that important information is lost when the imaginary part of A is discarded. Only performance of the fully complex AMG remains consistently good as β increases. In particular, note that complex AMG converges roughly twice as fast as real AMG. The complex AMG operator complexities remain steady for β away from 0; when $\beta = 0.25$, $c_A = 3.04$, while $c_A = 3.05$ for $\beta = 1.0$. Operator complexities for geometric multigrid are approximately 1.60 for all grids, while real AMG generates complexities that are similar to those of complex AMG for small β but that are somewhat smaller for larger β; when $\beta = 0.25$, $c_A = 2.63$, while $c_A = 2.06$ for $\beta = 1.0$. Thus, even with the performance advantage of real arithmetic over complex arithmetic, the complex AMG solver proposed here is more efficient than the use of a solver with real-valued transfer operators for these problems.

As β increases, the effect of the randomness in the definition of the covariant Laplacian becomes more pronounced. An interesting test problem arises when the covariant Laplacian appears in combination with a Helmholtz term,

$$-\sum_{\mu} D_{\mu}^{2}\psi(\mathbf{x}) + m\psi(\mathbf{x}),$$

where the coefficient, m, is chosen so that the matrices remain positive definite but match the conditioning of the usual Laplacian. Such a shift mimics the behavior of the mass term in the Dirac operator. To do this, we compute the maximum-magnitude eigenvalue, λ, of the matrix, M, obtained by taking $\sum_{\mu} D_{\mu}^{2}$ (so that the off-diagonal entries have positive sign) and setting the diagonal to zero. By Gerschgorin's theorem, we expect the largest eigenvalue of $-\sum_{\mu} D_{\mu}^{2}$ to be approximately $4 + \lambda \approx 8$, while the smallest should be roughly $4 - \lambda$, where $0 \leq \lambda \leq 4$. Then, m is chosen so that $\frac{8}{4-\lambda+m} = \frac{1}{h^2}$, i.e., $m = 8h^2 - (4 - \lambda)$. We then diagonally scale the matrix by $\frac{1}{4+m}$ so that it has constant unit diagonal.

Even for large h, the effect of such a shift on AMG performance can be dramatic. The eigenvector approximation criterion [4, 19] states that for AMG to be effective, each eigenvector of A must be approximated by something in the range of interpolation with accuracy proportional to its eigenvalue. For large eigenvalues of A, the shift by m has little effect on this approximation property. For the smallest eigenvalues, however, the shift by m has a significant effect and these modes may be very slow to be resolved by a simple AMG cycle, as very accurate interpolation is needed to complement the very slow performance of relaxation on the modes associated with the smallest eigenvalues of A. Table 5.4 shows some representative AMG convergence factors.

However, this shift is relatively significant for only a few modes of the matrix and, thus, the poor AMG performance is easily overcome through the use of a Krylov subspace accelerator. As A is Hermitian and positive definite (and the AMG cycle can easily also be made so), we consider here the performance of an AMG-preconditioned conjugate gradient algorithm. Figure 5.3 shows the convergence histories of geometric multigrid and AMG (both based on the real part and the complex AMG proposed

Table 5.4
Complex AMG convergence factors for the shifted covariant Laplacian, with variation in β and grid size.

	$\beta = 0.0$	$\beta = 0.25$	$\beta = 0.5$	$\beta = 0.75$	$\beta = 1.0$
65×65	0.104	0.988	0.993	0.993	0.990
129×129	0.143	0.998	0.997	0.997	0.998
257×257	0.166	0.9992	0.9996	0.9995	0.9993
513×513	0.231	0.99986	0.99987	0.99985	0.99988

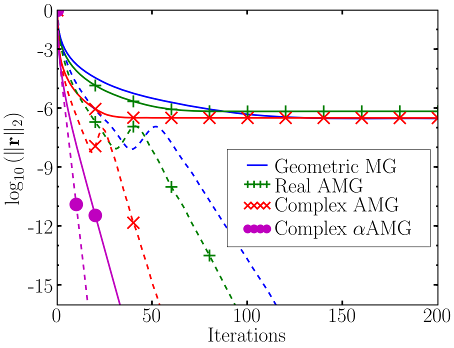


Fig. 5.3. *Convergence histories for geometric multigrid, AMG based on the real part of the matrix, complex AMG and adaptive complex AMG for the* 513×513 *shifted covariant Laplacian operator with* $\beta = 1.0$. *Solid lines indicate unaccelerated performance, while dashed lines indicate MG-PCG performance.*

here) for $\beta = 1.0$. For all three methods, slow (and stalling) convergence is seen for the unaccelerated solvers, while the MG-PCG combinations converge (relatively) quickly. Notice that the complex-AMG-PCG combination beats the real-AMG-PCG technique for convergence to any fixed tolerance by a factor of roughly 2.

An alternative to preconditioning to overcome the slowing down of convergence for the shifted covariant Laplacian is the use of adaptive multigrid techniques [9, 10]. In adaptive AMG [10], the approximation (2.4) used to collapse a strong connection between two fine-grid points, i and j, is replaced by one that takes into account the form of a representative algebraically smooth error exposed by adding an initial relaxation phase to the AMG setup algorithm. Thus, on each level in the AMG setup, we first relax on the homogeneous problem, $A\mathbf{u} = \mathbf{0}$, with a random initial guess for $\mathbf{u}$ to expose errors that relaxation is slow to resolve. This prototypical algebraically smooth error is then used in the definition of interpolation, in place of the AMG assumption that such errors vary slowly along strong connections. When this error is very different from the constant, the improvement in performance of adaptive AMG over classical AMG may be significant, as the classical AMG algorithm aims to satisfy the eigenvector approximation criteria for the constant vector only.

Figure 5.4 shows the algebraically smooth error found by performing 200 iterations of Gauss–Seidel relaxation on $A\mathbf{u} = \mathbf{0}$ (so that the error is well resolved), with a random initial guess for $\mathbf{u}$, on the shifted covariant Laplacian with $\beta = 1.0$ on a 65×65 grid. Thus, we expect a significant benefit of using adaptive AMG over the classical AMG assumption. Indeed, in Figure 5.3, the adaptive AMG convergence, both with and without PCG acceleration, is significantly better than that of any of the other methods. Table 5.5 shows adaptive AMG setup times and convergence factors

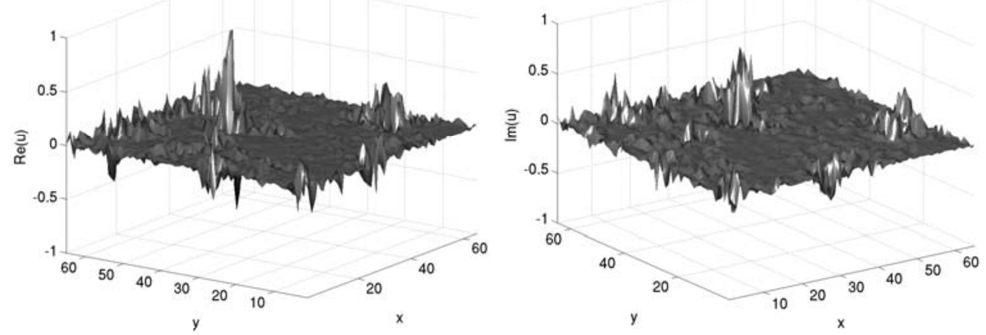


FIG. 5.4. *Algebraically smooth error for the shifted covariant Laplacian with* $\beta = 1.0$ *on a* 65×65 *mesh with the real part shown at left and the imaginary part shown at right.*

TABLE 5.5
Adaptive AMG setup times and convergence factors for the shifted covariant Laplacian.

	$\beta = 0.5$		$\beta = 0.75$		$\beta = 1.0$	
	t_{setup}	ρ	t_{setup}	ρ	t_{setup}	ρ
65×65	0.04 s	0.431	0.04 s	0.375	0.03 s	0.454
129×129	0.2 s	0.341	0.2 s	0.308	0.2 s	0.440
257×257	0.8 s	0.467	1.1 s	0.463	0.9 s	0.391
513×513	5.3 s	0.576	3.5 s	0.442	3.6 s	0.457

for several grid sizes and values of β.

The convergence factors in Table 5.5 do not appear to degrade as β increases and degrade only slightly with increase in problem size. It is not immediately clear if this degradation is due only to the increase in grid size or if it is related to the changes in the random sample taken for the gauge field on each grid. Setup times scale nearly with problem size, although a slight increase in the work needed (relative to problem size) for the adaptive AMG setup stage is required for each finer grid. In comparison, setup time for regular AMG on the 513×513 grid is 3.2 s, while approximately 0.36 s are required for a single V(1,1) cycle on that grid. Thus, the increase in cost of the adaptive AMG setup phase over the regular AMG setup phase (3.5-5.3 seconds versus 3.2 seconds) is roughly equivalent to the cost of one to six V-cycles, much less than the expected improvement offered in the adaptive AMG solve phase.

6. Conclusions. A natural extension of the algebraic multigrid method for complex-valued matrices is presented. Unlike previous extensions, our approach is completely algebraic in nature and relies on no special structure of the complex-valued matrix. Choices for the generalization are motivated by a combination of classical multigrid considerations and local Fourier analysis. Numerical results confirm the performance on simple model problems, realistic complex Helmholtz problems on unstructured meshes, and, in combination with Krylov acceleration or adaptive multigrid ideas, for ill-conditioned matrices based on covariant derivatives.

Acknowledgments. The authors would like to thank Domenico Lahaye for the problems considered in section 5.2 and James Brannick and Mike Clark for helpful discussion in defining the covariant Laplacian operators considered in section 5.3.

REFERENCES

[1] M. F. Adams, *Algebraic multigrid methods for direct frequency response analyses in solid mechanics*, Comput. Mech., 39 (2007), pp. 497–507.

[2] R. E. Alcouffe, A. Brandt, J. E. Dendy, and J. W. Painter, *The multigrid method for the diffusion equation with strongly discontinuous coefficients*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 430–454.

[3] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[4] A. Brandt, *Algebraic multigrid theory: The symmetric case*, Appl. Math. Comput., 19 (1986), pp. 23–56.

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

[6] A. Brandt, S. F. McCormick, and J. W. Ruge, *Algebraic Multigrid (AMG) for Automatic Multigrid Solution with Application to Geodetic Computations*, Tech. report, Institute for Computational Studies, Colorado State University, 1982.

[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, Lecture Notes in Comput. Sci. Engrg. 55, Springer, New York, 2007, pp. 505–512.

[8] J. Brannick, M. Brezina, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *An energy-based AMG coarsening strategy*, Numer. Linear Algebra Appl., 13 (2006), pp. 133–148.

[9] 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.

[10] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive algebraic multigrid*, SIAM J. Sci. Comput., 27 (2006), pp. 1261–1286.

[11] M. Brezina, T. Manteuffel, S. McCormick, J. Ruge, G. Sanders, and P. Vassilevski, *On Smooth Aggregation Multigrid for Nonsymmetric A*, Tech. report, University of Colorado, Boulder, 2006.

[12] C. Davies and P. Lepage, *Lattice QCD meets experiment in hadron physics*, AIP Conf. Proc., 717 (2004), pp. 615–624.

[13] D. Day and M. A. Heroux, *Solving complex-valued linear systems via equivalent real formulations*, SIAM J. Sci. Comput., 23 (2001), pp. 480–498.

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

[15] Y. A. Erlangga, C. W. Oosterlee, and C. Vuik, *A novel multigrid based preconditioner for heterogeneous Helmholtz problems*, SIAM J. Sci. Comput., 27 (2006), pp. 1471–1492.

[16] D. Lahaye, H. De Gersem, S. Vandewalle, and K. Hameyer, *Algebraic multigrid for complex symmetric systems*, IEEE Trans. Magn., 36 (2000), pp. 1535–1538.

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

[18] S. MacLachlan and Y. Saad, *A greedy strategy for coarse-grid selection*, SIAM J. Sci. Comput., 29 (2007), pp. 1825–1853.

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

[20] I. Montvay and G. Münster, *Quantum Fields on a Lattice*, Cambridge Monogr. Math. Phys., Cambridge University Press, Cambridge, 1994.

[21] S. Reitzinger, U. Schreiber, and U. van Rienen, *Algebraic multigrid for complex symmetric matrices and applications*, J. Comput. Appl. Math., 155 (2003), pp. 405–421.

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

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

[24] J. Smit, *Introduction to Quantum Fields on a Lattice*, Cambridge Lecture Notes Phys. 15, Cambridge University Press, Cambridge, 2002.

[25] K. Stüben and U. Trottenberg, *Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., Lecture Notes in Math. 960, Springer-Verlag, Berlin, 1982, pp. 1–176.

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

[27] P. Vaněk, J. Mandel, and M. Brezina, *Two-Level Algebraic Multigrid for the Helmholtz*

Problem, in Domain Decomposition Methods, 10 (Boulder, CO, 1997), Contemp. Math. 218, AMS, Providence, RI, 1998, pp. 349–356.

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

[29] R. S. Varga, *On recurring theorems on diagonal dominance*, Linear Algebra Appl., 13 (1976), pp. 1–9.

[30] R. Wienands and C. W. Oosterlee, *On three-grid Fourier analysis for multigrid*, SIAM J. Sci. Comput., 23 (2001), pp. 651–671.

[31] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.