SIAM J. Sci. Comput.
Vol. 0, No. 0, pp. 000–000

BOOTSTRAP AMG[*]

A. BRANDT[†], J. BRANNICK[‡], K. KAHL[§], and I. LIVSHITS[¶]

Abstract. We develop an algebraic multigrid (AMG) setup scheme based on the bootstrap framework for multiscale scientific computation. Our approach uses a weighted least squares definition of interpolation, based on a set of test vectors that are computed by a bootstrap setup cycle and then improved by a multigrid eigensolver and a local residual-based adaptive relaxation process. To emphasize the robustness, efficiency, and flexibility of the individual components of the proposed approach, we include extensive numerical results of the method applied to scalar elliptic partial differential equations discretized on structured meshes. As a first test problem, we consider the Laplace equation discretized on a uniform quadrilateral mesh, a problem for which multigrid is well understood. Then, we consider various more challenging variable coefficient systems coming from covariant finite-difference approximations of the two-dimensional gauge Laplacian system, a commonly used model problem in AMG algorithm development for linear systems arising in lattice field theory computations.

Key words. bootstrap algebraic multigrid, least squares interpolation, multigrid eigensolver, adaptive relaxation

AMS subject classification. 65F10, 65N55, 65F30

DOI. 10.1137/090752973

1. Introduction.

We develop bootstrap algebraic multigrid (BAMG) techniques for solving systems

$$Au = f, \tag{1.1}$$

where $A \in \mathbb{C}^{n\times n}$ is assumed to be Hermitian and positive definite. AMG approaches for solving (1.1) typically involve a stationary linear iterative method (smoother), applied to the fine-grid system, and a coarse-grid correction. The corresponding two-grid method gives rise to the error propagation operator

$$E_{TG} = (I - MA)(I - P(P^H AP)^{-1}P^H A)(I - MA), \tag{1.2}$$

where $P : \mathbb{C}^{n_c} \mapsto \mathbb{C}^n$ with $n_c < n$ is the interpolation operator, M is the approximate inverse of A that defines the smoother, and for any matrix $B \in \mathbb{C}^{n\times m}$, B^H denotes the conjugate transpose. A multigrid algorithm is then obtained by recursively solving the coarse-grid problem, involving $A_c = P^H AP$, using the two-grid method.

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section March 16, 2009; accepted for publication (in revised form) December 13, 2010; published electronically DATE.

http://www.siam.org/journals/sisc/x-x/75297.html

[†] Department of Mathematics, University of California Los Angeles, Los Angeles, CA 90095 (abrandt@math.ucla.edu).

[‡] Department of Mathematics, Pennsylvania State University, University Park, PA 16802 (brannick@psu.edu). This author's work was supported by the National Science Foundation under grants OCI-0749202 and DMS-810982.

[§] Fachbereich Mathematik und Naturwissenschaften, Bergische Universität Wuppertal, D-42097 Wuppertal, Germany (kkahl@math.uni-wuppertal.de). This author's work was supported by the Deutsche Forschungsgemeinschaft through the Collaborative Research Centre SFB-TR 55 "Hadron physics from Lattice QCD."

[¶] Department of Mathematical Sciences, Ball State University, Muncie, IN 47306 (ilivshits@bsu.edu). This author's work was supported by DOE subcontract B591416 and NSF DMS-0521097.

The efficiency of such an approach depends on the proper interplay between the smoother and the coarse-grid correction. Typically, the AMG smoothing operator, M, is fixed and the coarse-grid correction is formed to compensate for its deficiencies. The primary task in defining a multigrid method is then the selection of the sequence of interpolation operators. A general two-grid process for constructing P is described by the following generic algorithm:

1. Given the set of n variables on the fine grid, choose a set of n_c coarse variables such that $n_c < n$.
2. Choose a sparsity pattern for interpolation, $P \in \mathbb{C}^{n \times n_c}$.
3. Define the weights of the interpolation operator, i.e., the entries of P.

Classical AMG as originally introduced in [4, 5] and its widely used implementation by Ruge and Stüben [20] can be seen as important milestones in the development of such algebraic setup algorithms. The Ruge–Stüben algorithm exhibits optimal efficiency for many challenging problems, often substantially outperforming traditional iterative methods. However, the algorithm and, hence, its effectiveness depend on the following problem-specific properties:

- The notion of strength of connection used in coarsening variables and forming interpolation can be defined from the entries of the system matrix.
- The eigenvectors with small in absolute value eigenvalues (lowest eigenvectors) are locally smooth in directions of such strong connections.
- These lowest eigenvector(s) of the system matrix provide a sufficiently accurate local representation of the other low eigenvectors not effectively treated by the MG relaxation scheme.

For problems where all or some of these properties are violated, efficiency of Ruge–Stüben AMG deteriorates. This loss in efficiency is often due to the low accuracy of interpolation for vectors yielding small normalized residuals [1], i.e., vectors x such that $Ax \approx 0$. For simple pointwise relaxation schemes, such as the Gauss–Seidel iteration we use in our tests, these components coincide with the error not effectively reduced by the relaxation scheme and, hence, are often referred to as the algebraically smooth components of the error.

The bootstrap framework for multiscale scientific computation [3] provides general techniques for analysis, development, and practical application of robust multilevel methods. In the context of MG algorithms, the bootstrap process[1] allows the user to input any features of the problem at hand (e.g., nested grids and kernel components) and then use this knowledge to iteratively improve itself until optimal performance is achieved. The general setup algorithm used in this scheme incorporates several practical tools and measures derived from the evolving MG solver, including the following:

- compatible relaxation, used in identifying suitable smoothing schemes and coarse variable sets [2, 8, 12, 14];
- local weighted least squares approximations of a set of test vectors, used in a greedy algorithm to determine an algebraic distance-based measure of strength of connection [3] and to define AMG interpolation [14, 17, 19];
- the bootstrap cycling scheme, used to compute sufficiently accurate sets of test vectors [14, 15];

[1] Generally, the term bootstrap refers to the idea that a process could better evolve by improving the process used for its improvement (thus obtaining a compounding effect over time.) As a computing term, bootstrapping (from an old expression "pull oneself up by one's bootstraps") has been used since at least 1958 to refer to a technique by which a simple computer program activates a more complex system of programs that ultimately lead to a self-sustaining process that proceeds without external help from manually entered instructions.

- adaptive relaxation [2, 14] and almost zero modes [3], used to improve both the AMG setup cycle and the solver.

Many of the bootstrap ideas themselves are applicable to a wide range of multiscale problems in computational science. Our focus in this paper is on combining several of these techniques to develop a robust scheme for computing AMG interpolation.

A preliminary form of the bootstrap process for defining AMG interpolation first appeared in [5], where the use of relaxation applied to the homogeneous system to produce a single prototype to somehow define classical AMG interpolation was discussed. More recently, such bootstrap setup algorithms were developed for smoothed aggregation multigrid (adaptive SA; see [9]), element-free AMG (see [22, pp. 213–249]), and classical AMG (adaptive AMG, see [7, 10]). The main new ingredient in these more recent BAMG approaches is the idea to apply the current AMG solver to the homogeneous system to both test its performance and improve the prototypes used in computing the sequence of AMG interpolation operators.

The algorithm we consider here combines a BAMG cycle with a weighted least squares form of interpolation. An additional feature unique to our approach is the use of an MG eigensolver derived from the existing MG structure to enhance the prototypes needed in the definition of least squares interpolation. We mention that similar MG techniques for using eigenvectors to define AMG interpolation were roughly outlined in [18, 20]. In another related work [11], an MG eigensolver was developed to compute an initial SA hierarchy, after which it is abandoned and the usual adaptive SA setup process is invoked.

An outline of the remainder of this paper is as follows. First, in section 2, we present the weighted least squares process for computing interpolation and the idea of adaptive relaxation. In addition, we derive sufficient conditions guaranteeing the uniqueness of the solution to the least squares problem and compute an explicit form of the minimizer. Our approach for computing the set of test vectors using bootstrap techniques is described in section 3. Then, in section 4, we present results of the method applied to the scalar Laplace equation discretized on a uniform mesh and several challenging variable coefficient problems. We end with concluding remarks in section 5.

2. Least squares interpolation.

The basic idea of the least squares (LS) interpolation approach is to approximate a set of test vectors, $\mathcal{V} = \{v^{(1)}, \ldots, v^{(k)}\} \subset \mathbb{C}^n$, minimizing the interpolation error for these vectors in an LS sense. In the context considered here, namely, applying the LS process to construct a classical AMG form of interpolation, each row of P, denoted by p_i, is defined as the minimizer of a local LS functional: For each $i \in \mathcal{F}$ find p_i such that

$$\mathcal{L}(p_i) = \sum_{\kappa=1}^{k} \omega_\kappa \left(v_{\{i\}}^{(\kappa)} - \sum_{j \in \mathcal{C}_i} (p_i)_j \, v_{\{j\}}^{(\kappa)} \right)^2 \mapsto \min, \tag{2.1}$$

where $\mathcal{C}_i \subset C$ with $\mathcal{C}$ and $\mathcal{F} = \Omega \setminus \mathcal{C}$ denoting the coarse-grid and fine-grid variables, respectively. Here, the notation $v_{\tilde{\Omega}}$ denotes the canonical restriction of the vector v to the set $\tilde{\Omega} \subset \Omega := \{1, .., n\}$. In the definition of $\mathcal{L}$ in (2.1), for example, $v_{\{i\}}$ is simply the ith entry of v. Similarly, we can define the canonical restriction of a matrix, $V = \left(v^{(1)} \cdots v^{(k)}\right)$, to a set $\tilde{\Omega}$:

$$V_{\tilde{\Omega}} = \begin{pmatrix} v_{\tilde{\Omega}}^{(1)} & \cdots & v_{\tilde{\Omega}}^{(k)} \end{pmatrix},$$

where $V_{\tilde{\Omega}} \in \mathbb{C}^{|\tilde{\Omega}| \times k}$. The weights $\omega_\kappa > 0$ can be chosen to reflect the energy (e.g., in A-norm $\|v\|_A = \langle Av, v\rangle$) of the test vectors. We give our specific choice in the numerical experiments section.

2.1. Uniqueness of the solution to the local LS problem and an explicit form of its minimizer. To derive conditions on the uniqueness of the solution to minimization problem (2.1) and compute an explicit form of the minimizer, we consider a classical linear algebra formulation of the LS problem. Let

$$W = \begin{pmatrix} \omega_1 & & \\ & \ddots & \\ & & \omega_k \end{pmatrix} \in \mathbb{C}^{k\times k}, \qquad V = \begin{pmatrix} v^{(1)} & \cdots & v^{(k)} \end{pmatrix} \in \mathbb{C}^{n\times k},$$

and $V_{\tilde{\Omega}} = (v_{\tilde{\Omega}}^{(1)} \quad \cdots \quad v_{\tilde{\Omega}}^{(k)})$. Then,

$$\sum_{\kappa=1}^{k} \omega_\kappa \left(v_{\{i\}}^{(\kappa)} - \sum_{j\in\mathcal{C}_i} (p_i)_j v_{\{j\}}^{(\kappa)} \right)^2 = \|V_{\{i\}} W^{\frac{1}{2}} - p_i V_{\mathcal{C}_i} W^{\frac{1}{2}}\|_2^2,$$

where the weights of the individual terms in the LS functional are now represented by a scaling defined by the matrix W. An equivalent LS formulation of (2.1) is then

$$\mathcal{L}(p_i) = \|V_{\{i\}} W^{\frac{1}{2}} - p_i V_{\mathcal{C}_i} W^{\frac{1}{2}}\|_2^2 \to \min. \tag{2.2}$$

The minimizer of $\mathcal{L}$ in (2.2) and necessary conditions for its uniqueness are now easy to compute. The derivative of the LS functional $\mathcal{L}$ with respect to $(p_i)_j$ is

$$\frac{\partial}{\partial (p_i)_j} \mathcal{L}(p_i) = 2 \sum_{\kappa=1}^{k} \omega_\kappa \left(v_{\{i\}}^{(\kappa)} - p_i v_{\mathcal{C}_i}^{(\kappa)} \right) \left(v_{\mathcal{C}_i}^{(\kappa)} \right)_j .$$

Setting $\nabla \mathcal{L}(p_i) = 0$ yields

$$p_i V_{\mathcal{C}_i} W V_{\mathcal{C}_i}^H = V_{\{i\}} W V_{\mathcal{C}_i}^H .$$

Thus, if $\operatorname{rank}(V_{\mathcal{C}_i} W^{\frac{1}{2}}) = |\mathcal{C}_i|$, then $V_{\mathcal{C}_i} W V_{\mathcal{C}_i}^H$ is nonsingular and the solution to the LS minimization problem is uniquely defined by

$$p_i = V_{\{i\}} W V_{\mathcal{C}_i}^H \left(V_{\mathcal{C}_i} W V_{\mathcal{C}_i}^H \right)^{-1} .$$

We note that if the restricted vectors $V_{\mathcal{C}_i}$ form a basis for the local linear space $\mathbb{C}^{n_i}$, $n_i = |C_i|$, then the solutions to the local LS minimization problems are unique. This in turn suggests setting a lower bound on the number of vectors, k, used in the LS fit:

$$k \geq \max_{i\in\mathcal{F}} |\mathcal{C}_i| =: c.$$

Further, as we show numerically in section 4, the accuracy of the LS interpolation operator and, hence, the performance of the resulting solver generally improve with increasing k, up to some value proportional to c. Our numerical experience suggests that the number of test vectors need not be larger than $2c$ to obtain a sufficiently accurate P. In fact, these sets of interpolation points can often be adequately chosen by natural considerations. For example, they can be chosen as the sets of geometrical neighbors with i in their convex hull. If a chosen set is inadequate, the LS procedure will show *bad fitness* (large interpolation errors, i.e., values of the LS functional) and the set must then be improved. The LS procedure can also be used to detect variables in the sets $\mathcal{C}_i$ that can be discarded without significant accuracy loss. Thus, this approach allows creating interpolation with whatever needed accuracy, which is *as sparse as possible.* Note that it is also possible to adjust the number of test vectors on the fly if rank deficiency of any of the operators $V_{\mathcal{C}_i}$ is detected. Similarly, many rows of P will typically have fewer than c nonzero entries, that is, $|\mathcal{C}_i| < c$. In such cases, fewer prototypes may be used in the LS process. We do not pursue this idea in this paper because it is not expected to result in a significant reduction in setup costs, and further, as our numerical results show, the quality of the solver generally improves with an increased number of prototypes.

2.2. Equivalence of adaptive relaxation and a modified LS functional. A main assumption of classical AMG is that the relaxation scheme used in the MG algorithm efficiently reduces the residual when applied to the current approximation. This assumption is in fact central to the definition of classical AMG interpolation, which is derived by setting the local (pointwise) residual to zero. In [2], the idea of applying additional local relaxations to the equations $i \in \mathcal{F}$ for which the corresponding value of the residual is large was proposed as a possible approach for improving the performance of classical AMG for certain problems (e.g., problems with singularities). Assuming $a_{ii} \neq 0$, a Jacobi version of this iteration reads as

$$v_{\{i\}}^{(\kappa)} = v_{\{i\}}^{(\kappa)} - \frac{1}{a_{ii}} r_{\{i\}}^{(\kappa)}, \tag{2.3}$$

with $r^{(\kappa)} = Av^{(\kappa)}$. Similar approaches can also be employed for more general relaxation schemes.

In [14], it was observed that the implicit application of the local Jacobi relaxation in (2.3) to the prototypes used in the LS definition of interpolation is equivalent to an operator-induced form of LS interpolation constructed using an element-free AMG type approach (see [22]). This equivalence of the LS approach with an additional local relaxation step was also formulated and discussed in a slightly different scope in [17], where it was defined as a residual-based LS fit,

$$\mathcal{L}(p_i) = \sum_{\kappa=1}^{k} \omega_\kappa \left(v_{\{i\}}^{(\kappa)} - \frac{1}{a_{ii}} r_{\{i\}}^{(\kappa)} - \sum_{j\in\mathcal{C}_i} (p_i)_j v_{\{j\}}^{(\kappa)} \right)^2 \mapsto \min, \tag{2.4}$$

which in turn was shown to be consistent with a classical AMG operator-induced form of LS interpolation. We mention that the work in [17] focuses on defining different versions of the LS interpolation operators using a set of relaxed test vectors; it does not address the question of how to use the bootstrap process to compute these vectors. Developing the bootstrap setup scheme for defining LS interpolation is the main focus and contribution of the work we present here.

Another interesting observation regarding the above residual form of the LS functional is that the element-free AMG and classical AMG forms of LS interpolation considered in [14, 17] differ from the original definition only when the test vectors with large weights, $\omega_{k_\star}$, are not sufficiently accurate; i.e., the residuals they produce are not uniformly close to zero. In fact, our idea of applying adaptive relaxation to the test vector with largest weight (2.4) and then only to the equations with large relative values of the residual was motivated by these observations. We include results and additional discussion of both approaches in the numerical experiments section.

3. The bootstrap algorithm.

In its simplest form, the bootstrap process for computing the test vectors used in constructing the LS interpolation operators proceeds by applying relaxation to the homogeneous system,

$$A_l x_l = 0, \tag{3.1}$$

on each grid, where $l = 0, \ldots, L-1$; assuming that a priori knowledge of the algebraically smooth error is not available, these vectors are initialized randomly on the finest grid, whereas on all coarser grids they are defined by restricting the test vectors computed on the previous finer grid. Given interpolation, the coarse-grid operators are computed using the variational definition. Once an initial MG hierarchy has been computed, the current sets of prototypes are further enhanced on all grids using the existing multigrid structure. Specifically, the given hierarchy is used to formulate a multigrid eigensolver which is then applied to an appropriately chosen generalized eigenproblem to compute additional test vectors. This overall process is then repeated with the current AMG method replacing relaxation as the solver for the homogeneous systems in (3.1).

Several subtle details must be addressed when formulating the multigrid eigensolver (MGE), namely, (1) the specific formulation for the MGE hierarchy as well as the sorting and filtering of the coarsest-grid eigenvectors within the cycle; (2) deciding on an efficient cycling strategy when integrating the MGE into the BAMG process; and (3) measuring and improving the accuracy of the test vectors computed by the MGE (eigenapproximations) as they are transferred to increasingly finer grids. We describe our formulation of the MGE next.

3.1. Defining the MGE hierarchy: The generalized eigenvalue problem.

Given the computed BAMG hierarchy of operators $A = A_0, A_1, \ldots, A_L$ and their corresponding interpolation operators P_{l+1}^{l}, $l = 0, \ldots, L-1$, define the composite interpolation operators

$$P_l = P_1^0 \cdots P_l^{l-1}, \quad l = 1, \ldots, L \tag{3.2}$$

and the sequence of subspaces of $\mathbb{C}^n$ spanned by their columns as

$$\mathbb{C}^n \supset \operatorname{range}(P_1) \supset \cdots \supset \operatorname{range}(P_L).$$

For any vector $x_l \in \mathbb{C}^{n_l}$ we then have

$$\langle x_l, x_l \rangle_{A_l} = \langle P_l x_l, P_l x_l \rangle_A,$$

where n_l is the problem size on grid l. Defining $T_l = P_l^H P_l$, it follows that

$$\frac{\langle x_l, x_l \rangle_{A_l}}{\langle x_l, x_l \rangle_{T_l}} = \frac{\langle P_l x_l, P_l x_l \rangle_A}{\langle P_l x_l, P_l x_l \rangle_2}, \tag{3.3}$$

implying that on any grid l, given a vector $w_l \in \mathbb{C}^{n_l}$ and $\lambda_l \in \mathbb{R}$, such that

$$A_l w_l = \lambda_l T_l w_l, \tag{3.4}$$

we have

$$\mathrm{RQ}(P_l w_l) = \frac{\langle P_l w_l, P_l w_l\rangle_A}{\langle P_l w_l, P_l w_l\rangle_2} = \lambda_l. \tag{3.5}$$

The result follows from (3.3) applied to w_l, the eigenvector of the generalized eigenvalue problem (3.4), and the relations $A_l = P_l^H A P_l$ and $T_l = P_l^H P_l$. Note that $\mathrm{RQ}(\cdot)$ is simply the Rayleigh quotient and, hence, (3.5) relates the eigenpairs on different grids and can be used to define an MGE.

Our algorithm begins by computing the k_e eigenvectors with the smallest eigenvalues of the coarsest-grid operator

$$\mathcal{W}_L = \{w_L^{(\kappa)} \mid A_L w_L^{(\kappa)} = \lambda_L^{(\kappa)} T_L w_L^{(\kappa)},\ \lambda_L^{(\kappa)} \in \mathbb{R},\ \kappa = 1,\ldots,k_e\}.$$

Since the size of A_L is small, these eigenpairs, $\left(w_L^{(\kappa)}, \lambda_L^{(\kappa)}\right)$, are computed directly. Then, on any given grid l, the existing interpolation operator P_l^{l-1} is used to transfer these vectors to the next finer grid. We note that if w_l is a solution to (3.4) on grid l, then

$$\left(P_l^{l-1}\right)^H A_{l-1} P_l^{l-1} w_l = \lambda_l \left(P_l^{l-1}\right)^H T_{l-1} P_l^{l-1} w_l,$$

and so $w_{l-1} = P_l^{l-1} w_l$ is an approximate solution to the generalized eigenvalue problem. Next, a smoothing iteration is applied to the homogeneous problem on grid $l-1$:

$$(A_{l-1} - \lambda_{l-1} T_{l-1})\, w_{l-1} = 0, \quad \lambda_{l-1} = \lambda_l. \tag{3.6}$$

Then, the approximation to λ_{l-1} is recomputed:

$$\lambda_{l-1} = \frac{\langle A_{l-1} w_{l-1}, w_{l-1}\rangle_2}{\langle T_{l-1} w_{l-1}, w_{l-1}\rangle_2}.$$

We note that this procedure resembles an inverse Rayleigh quotient iteration found in eigenvalue computations (see [23]), with inversion replaced by several relaxation steps. Algorithm 1 describes one variant of the MGE process for enriching the sets of test vectors.

3.2. Integrating the MGE and BAMG processes: Cycling strategies. Combining the bootstrap cycle with the MGE can be done using a variety of cycling strategies. The two types of cycling schemes we consider are outlined in Figure 3.1. The top plot is a visualization of two successive iterations of a V-cycle scheme, which we refer to as a double V-cycle (or V^2-cycle). In general, a V^μ-cycle denotes an algorithm that uses μ such V-cycles. We note that this scheme makes use of the MGE as defined by Algorithm 1, which does not update interpolation until it reaches the finest grid.

The plot in the bottom of the figure outlines our second approach, in which the approximations produced by the MGE on grid l are used to recompute the hierarchy at each of the coarser grids, $l+1, l+2, \ldots, L$, before advancing to the next finer grid $l-1$. Note that it may not be necessary to recompute the hierarchy at each of the intermediate grids; in practice we recompute P and update the hierarchy only when relaxation applied to the interpolated eigenvector approximations significantly reduces at least one of the Rayleigh quotients (3.5) associated with these approximations by some prescribed tolerance.

Algorithm 1. {Multigrid eigensolver}

```
for $l = L, \dots, 1$ do
  if the current grid is the coarsest grid $L$ then
    Take $\mathcal{W}_L = \{w_L^{(\kappa)} \mid A_L w_L^{(\kappa)} = \lambda_L^{(\kappa)} T_L w_L^{(\kappa)}, \kappa = 1, \dots, k_e\}$
  else
    Given $\mathcal{W}_{l+1}$ and $P_{l+1}^l$ from the initial setup
    $w_l^{(\kappa)} = P_{l+1}^l w_{l+1}^{(\kappa)}, \quad \lambda_l^{(\kappa)} = \lambda_{l+1}^{(\kappa)}, \quad \kappa = 1, \dots, k_e$
    for $\kappa = 1, \dots, k_e$ do
      Relax on $\left(A_l - \lambda_l^{(\kappa)} T_l\right) w_l^{(\kappa)} = 0$
      Calculate $\lambda_l^{(\kappa)} = \frac{\langle A_l w_l^{(\kappa)}, w_l^{(\kappa)} \rangle_2}{\langle T_l w_l^{(\kappa)}, w_l^{(\kappa)} \rangle_2}$
    end for
  end if
end for
```

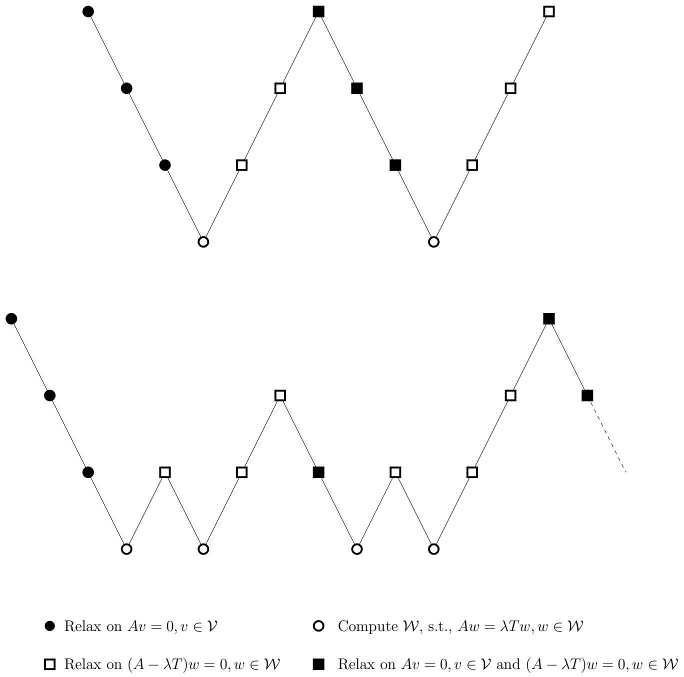

● Relax on $Av = 0, v \in \mathcal{V}$
○ Compute $\mathcal{W}$, s.t., $Aw = \lambda T w, w \in \mathcal{W}$
□ Relax on $(A - \lambda T)w = 0, w \in \mathcal{W}$
■ Relax on $Av = 0, v \in \mathcal{V}$ and $(A - \lambda T)w = 0, w \in \mathcal{W}$

Fig. 3.1. *Bootstrap AMG V^2-cycle and W-cycle setup schemes.*

3.3. Updating the MG eigendecomposition. Consider the generalized eigenvalue problem

$$A_j w_j = \lambda_j T_j w_j,$$

on two subsequent grids, $j = l,\ l-1$, with λ_l denoting the Rayleigh quotient of a given approximation of an eigenvector on the coarser grid and λ_{l-1} denoting the Rayleigh quotient of the vector obtained by applying relaxation to this vector interpolated to the next finer one. Define the *eigenvalue approximation measure* $\tau_\lambda^{(l,l-1)}$ by

$$\tau_\lambda^{(l,l-1)} = \frac{|\lambda_l - \lambda_{l-1}|}{|\lambda_{l-1}|}. \tag{3.7}$$

At any stage of the MGE iteration, a large value of $\tau_\lambda^{(l,l-1)}$ indicates that the hierarchy should be recomputed to incorporate this relaxed eigenvector approximation. Otherwise, relaxation has not significantly changed this vector, and it is accurately represented by the existing interpolation operator P_l^{l-1}. In this way, the MGE serves as a technique for efficiently identifying components that must be interpolated accurately (e.g., the low modes of A) and for determining if the current P approximates them sufficiently well.

4. Numerical results. In this section, we present numerical tests of our BAMG-MGE setup algorithm applied to a series of test problems. We consider isotropic systems defined on two-dimensional (2D) equidistant quadrilateral meshes. Fixing the coarse grids and the sparsity pattern of interpolation, we study the performance of the LS and MGE techniques for computing the entries in P. We begin with tests for the Laplace operator discretized using finite elements (FE Laplace) and finite differences (FD Laplace). We then transition to a slightly more difficult test problem of symmetric diagonal scalings of the Laplace operator. Then, we proceed to tests of our method applied to the gauge Laplacian system (see Appendix A).

In all tests, the sets of coarse variables are defined by full coarsening; that is, the coarse grids are obtained by doubling the mesh spacing in each spatial dimension of the related finer mesh; the coarsening is continued until the problem size on the coarsest grid is 7×7 or 8×8, depending on the size of the problem on the finest mesh. We limit interpolation to the nearest neighbors (in terms of the graph of the matrix), and the maximal number of interpolatory points for each $i \in \mathcal{F}$ is thus bounded by four for the problems we consider (see Figure 4.1).

We use the weighted LS approach to define the entries of the interpolation operators, with the weights defined by

$$\omega_\kappa = \frac{\langle T_l v^{(\kappa)}, v^{(\kappa)} \rangle}{\langle A_l v^{(\kappa)}, v^{(\kappa)} \rangle}, \tag{4.1}$$

and test the original LS definition of interpolation (LS interpolation) as well as the approach that applies additional adaptive local relaxations to the test vectors prior to computing the interpolation operator (LSR interpolation). For the results reported in Table 4.10, we apply the local relaxation scheme (2.3) to every test vector; for all other tests we apply local relaxation only to the single test vector with the largest weight, $\omega_{\max}$, and then only to 20% of the $\mathcal{F}$-points for which the associated value of the residual is largest in absolute value.

Let $k_r = |\mathcal{V}|$ denote the number of test vectors computed using relaxation in the bootstrap cycle, let $k_e = |\mathcal{W}|$ denote the number of additional eigenvector approximations computed by the MGE, and let η be the number of Gauss–Seidel (GS) iterations used in the BAMG cycle and MGE to compute them. The LS and LSR forms of interpolation are then computed on each level using the combined sets of $k = k_r + k_e$ test vectors. On the finest grid, the vectors, $v^{(1)}, \ldots, v^{(k_r)}$, used to initialize the bootstrap process are generated randomly with a normal distribution with expectation zero and variance one, $N(0,1)$; on all other grids they are defined by restricting the relaxed test vectors from the related finer grid.

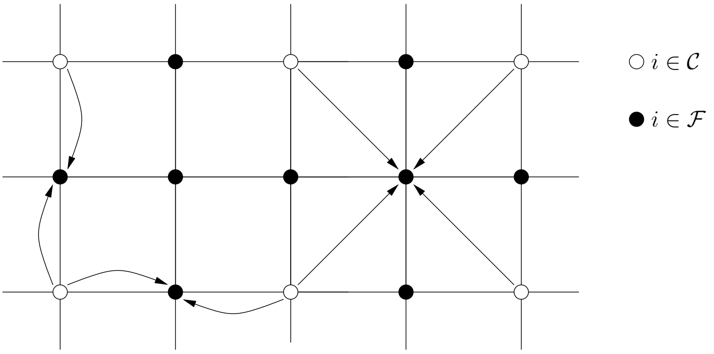


FIG. 4.1. *Interpolation relations for our choice of full-coarsening for* $i \in \mathcal{F} \setminus \mathcal{C}$.

We use a $V(2,2)$ MG solver with GS relaxation for all tests. The reported estimates of the asymptotic convergence rates are computed as follows:

$$\rho = \frac{\|e^{\nu}\|_A}{\|e^{\nu-1}\|_A},$$

where e^{ν} is the error after ν MG iterations. The solver terminates if the method reduces the initial error by a factor 10^8 or if $\nu = 100$ iterations are applied and the method fails to converge to this tolerance.

4.1. Laplace's equation. We start the numerical experiments by applying the BAMG cycle and MGE to the 2D Laplace equation

$$\begin{aligned} -u_{xx} - u_{yy} &= 0, \quad (x,y) \in (0,1)^2, \\ u(x,y) &= 0, \quad (x,y) \in \Gamma, \end{aligned}$$

discretized using bilinear finite elements (see [21]) on an $(N-1) \times (N-1)$ equidistant quadrilateral mesh. The corresponding discretization yields the symmetric and positive definite operator A given by the stencil

$$A = -\frac{1}{3} \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

For this problem our choices of full coarsening and sparsity pattern of interpolation yield variational coarse-grid operators with at most nine nonzero entries per row, implying that the sparsity structure of the finest-grid operator A is preserved on all grids.

The results reported in Table 4.1 are for the LS- and LSR-based two-grid methods applied to the FE Laplace system on a 63×63 grid. Overall, we observe that increasing the number of test vectors, k_r, used in defining P and increasing the number of smoothing iterations, η, used to compute them both improve the performance (reduce the convergence rates) of the resulting two-grid solvers. Further, the results for the LSR scheme are consistently better than the results for the LS approach.

In Table 4.2, we report results obtained by applying a two-grid method to the FE Laplace system for different problem sizes. The solver is constructed using the LS and LSR formulations for defining P with a fixed numbers of test vectors $k_r = 8$ and smoothing iterations $\eta = 4$ independent of the choice of problem size. Here, we see

Table 4.1

Asymptotic convergence rate estimates, ρ, of the LS- (LSR)-based two-grid methods applied to the FE Laplace problem. The solver is constructed using various choices of η and k_r.

η \ k_r	6	7	8	10	12
1	.975 (.962)	.973 (.958)	.971 (.955)	.967 (.949)	.963 (.944)
2	.934 (.866)	.916 (.827)	.904 (.806)	.878 (.786)	.861 (.759)
3	.832 (.679)	.803 (.637)	.775 (.606)	.734 (.566)	.709 (.535)
4	.735 (.524)	.681 (.453)	.648 (.403)	.608 (.269)	.575 (.316)
5	.625 (.374)	.553 (.326)	.518 (.287)	.484 (.165)	.457 (.138)
6	.522 (.294)	.451 (.242)	.425 (.205)	.381 (.154)	.305 (.136)
7	.427 (.227)	.364 (.186)	.346 (.155)	.268 (.089)	.257 (.098)
8	.372 (.169)	.308 (.140)	.287 (.123)	.238 (.054)	.216 (.061)

Table 4.2

Asymptotic convergence rate estimates, ρ, of the LS- (LSR)-based two-grid methods applied to the FE Laplace problem. The solver is constructed using the LS (LSR) schemes in a V-cycle setup with $\eta = 4$ and $k_r = 8$.

N	31	63	127	255	511
ρ	.267 (.075)	.648 (.403)	.886 (.758)	.966 (.917)	.990 (.977)

that the convergence rates of both the LS and LSR based two-grid methods increase as the size of the problem is increased, suggesting that applying a constant number of relaxation steps to compute a fixed number of test vectors does not produce a sufficiently accurate local representation of the algebraically smooth error for defining LS interpolation.

In Table 4.3, we present results for the two-grid method obtained by adding the constant vector, $\mathbf{1}$, to $\mathcal{V}$, again for a fixed number of test vectors $k_r = 7+1$ and $\eta = 4$ relaxation steps used to compute them. Here, we see that including the constant vector in $\mathcal{V}$ significantly improves the LS- and LSR-based two-grid solvers. In fact, the method obtained using the LSR approach performs optimally for the problem sizes considered. This demonstrates the flexibility of the LS formulation and its ability to incorporate a priori knowledge of the low modes of the problem at hand. It also motivates the use of the MGE in the BAMG process for computing test vectors, as a way to improve their approximation of low eigenmodes.

Table 4.3

Asymptotic convergence rate estimates, ρ, of the LS- (LSR)-based two-grid methods applied to the FE Laplace problem. The solver is constructed using the LS (LSR) schemes using a V-cycle setup with $\eta = 4$ GS iterations to compute the seven initially random test vectors; the additional test vector is the constant vector chosen a priori.

N	31	63	127	255	511
ρ	.089 (.039)	.121 (.040)	.130 (.042)	.148 (.042)	.150 (.043)

Next, we consider using the V^2-cycle setup (see Figure 3.1). We apply $\eta = 4$ iterations of the smoother to $k_r = 8$ initially random test vectors to generate the MG hierarchy and $\eta = 4$ relaxation steps to the $k_e = 8$ additional test vectors generated by the MGE. As the results reported in Table 4.4 show, using the MGE to enhance the test vectors consistently improves the performance of the resulting solvers when compared to the results reported in Table 4.2, especially for the LSR scheme. Here, the LSR approach yields an optimal method, whereas for the LS scheme, as the problem size goes from $N = 255$ to $N = 511$, the convergence rate of the solver

TABLE 4.4

Asymptotic convergence rate estimates, ρ, of the LS- (LSR)-based two-grid methods applied to the FE Laplace problem. The solver is constructed using a V^2-cycle setup with $\eta = 4$ and $k_r = k_e = 8$.

N	31	63	127	255	511
ρ	.041 (.038)	.062 (.041)	.075 (.043)	.125 (.043)	.971 (.043)

increases substantially—from $\rho \approx .125$ to $\rho \approx .971$. These results in turn suggest that the global weights, ω_κ, used in computing interpolation, bias the LS fit to some inadequate local representations of the algebraically smooth error, ultimately leading to a poor choice of interpolation for the associated fine-grid variables. The LSR scheme is, however, able to compensate for this in a very efficient manner. An alternative approach would be to define the weights used in the LS fit locally (i.e., for each $i \in \mathcal{F}$)—we mention that although we have studied this idea extensively, we have not yet derived an effective strategy for defining local weights.

To further illustrate the efficacy of the MGE when combined with the LSR form of P for this problem, we report the eigenvalue approximation measures $\tau_\lambda^{(L,1)}$ defined in (3.7), computed using the eight smallest eigenvalues of the coarsest-grid operator and the associated Rayleigh quotients of the finest-grid test vectors generated using the V^2-cycle setup algorithm. Additionally, we report the two-norms of the differences between the eight eigenvectors with the smallest eigenvalues of the coarsest-grid system interpolated to the finest grid using the composite interpolation operators defined in (3.2) and the eigenvectors of the finest-grid system computed directly. The results are reported in Table 4.5. Here, we see that the eigenvalue approximation measures are uniformly small for the $k_e = 8$ computed eigenvector approximations and further that the eigenvector approximations interpolated form the coarsest grid to the finest one approximate well the targeted eigenvectors of the finest grid system, i.e., the eigenvectors associated with the smallest eigenvalues of A. We mention further that the eigenvector approximation measures consistently detect the accuracy of the eigenvector approximations computed using the MGE.

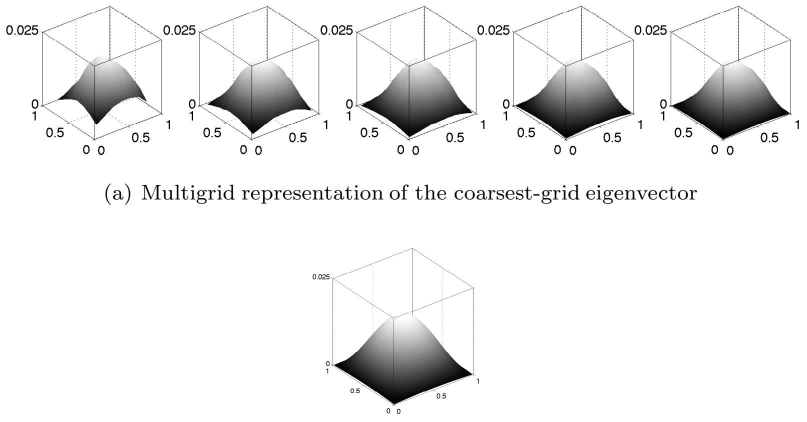
TABLE 4.5

Relative eigenvalue approximation measures $\tau_{\lambda_i}^{(L,1)}$ and eigenvector approximation estimates of the eight smallest eigenvalues for the FE Laplace problem, computed within a V^2-cycle setup with $\eta = 4$ and $k_r = k_e = 8$.

i	1	2	3	4	5	6	7	8
$\tau_{\lambda_i}^{(L,1)}$	.0153	.0296	.0302	.0381	.0792	.0829	.0736	.0762
$\|P_L v_i^L - v_i\|_2$	.0099	.0228	.0225	.0332	.0549	.0544	.0581	.0609

Next, we provide plots of the eigenvector approximations computed using a V-cycle setup algorithm with $\eta = 4$ and $k_r = k_e = 8$ and of the associated eigenvectors of the finest grid operator computed directly. Figure 4.2(a) contains plots of the eigenvector with the smallest eigenvalue on each level of the MGE computed using a single V-cycle setup, and Figure 4.2(b) contains a plot of the eigenvector with the smallest eigenvalue of the system matrix on the finest grid computed directly. The plots demonstrate the ability of the MGE (built using smoothed random test vectors) to recover the smallest eigenvector of the finest-grid FE Laplace operator.

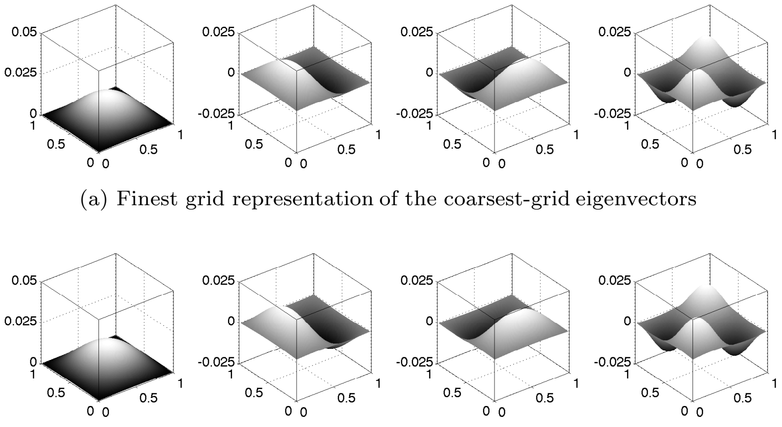
In addition, we provide plots comparing the eigenvector approximations on the finest grid computed using the same V^2-cycle setup and the eigenvectors of the finest-



(a) Multigrid representation of the coarsest-grid eigenvector

(b) Finest grid

FIG. 4.2. *Comparison of the eigenvector corresponding to the smallest eigenvalue in* (a) *computed in a V-cycle setup with $\eta = 4$ and $k_r = k_e = 8$ and the associated finest grid eigenvector in* (b) *corresponding to the smallest eigenvalue of the FE Laplace problem computed directly.*



(a) Finest grid representation of the coarsest-grid eigenvectors

(b) Finest grid

FIG. 4.3. *Visualization of the finest-grid eigenvector approximations of the four smallest eigenvalues computed using a V^2-cycle setup with $\eta = 4$ and $k_r = 8$ in* (a) *and the associated exact finest-grid eigenvectors corresponding to the four smallest eigenvalues in* (b).

grid operator computed directly. Figure 4.3(a) contains plots of the finest-grid eigenvector approximations for the four smallest eigenvalues computed using a V^2-cycle setup, and Figure 4.3(b) contains the plots of the eigenvectors of the finest-grid operator with the four smallest eigenvalues computed directly. Again, the plots illustrate that the MGE is able to recover the low eigenvectors of the fine-grid system.

We now consider tests of the W-cycle setup, illustrated at the bottom of Figure 3.1. In Table 4.6, we present results for this scheme again with $k_r = k_e = 8$. We notice a marked improvement in the performance of the LS-based solver constructed using a W-cycle setup over that of the solver constructed using a V-cycle setup and note that the LSR scheme again scales optimally with problem size.

TABLE 4.6

Asymptotic convergence rate estimates, ρ, of the LS- (LSR)-based MG solvers applied to the FE Laplace problem. The solver is constructed using a W-cycle setup (see Figure 3.1) with $\eta = 4$ and $k_r = k_e = 8$.

N	31	63	127	255	511
ρ	.042 (.038)	.062 (.041)	.073 (.043)	.130 (.043)	.161 (.044)

The last result we present for the FE Laplace system is related to the computational complexity of the BAMG-MGE setup. In Figure 4.4, we show the time, t, spent in the setup phase as a function of the number of the finest-grid variables, N^2. The numerical tests were performed using an 2.53GHz Intel Core 2 Duo processor with 4GB 1066MHz DDR3 SDRAM. The results suggest that the computational complexity for both setup cycling strategies is nearly optimal (grows linearly with the problem size) when applied to this problem.

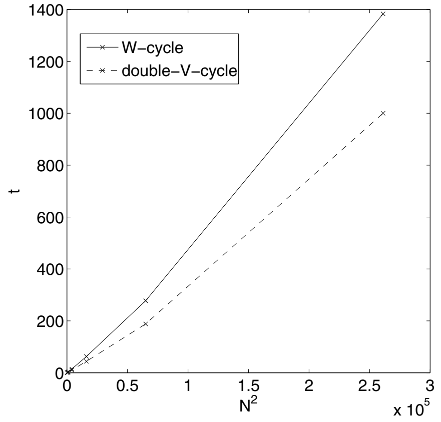


FIG. 4.4. *Total wall clock times for the V^2- and W-cycle setup algorithms with $\eta = 4$ and $k_r = k_e = 8$ applied to the FE Laplace problem versus the problem size N^2.*

4.2. The gauge Laplacian system. In this section, we consider solving the 2D gauge Laplacian system [6, 13, 16] with periodic boundary conditions, a problem that is typically used for testing potential AMG algorithms for solving the Dirac equation in more general lattice gauge theories, e.g., quantum chromodynamics (QCD). A detailed discussion of the GL system and its spectral properties is provided in Appendix A.

The 2D GL can be viewed as a stochastic variant of the standard Laplace operator discretized using central finite differences. Specifically, using stencil notation, the GL operator is given by

$$A(\mathcal{U}) = \begin{pmatrix} & -\overline{U_y^z} & \\ -U_x^{z-e_x} & 4+m & -\overline{U_x^z} \\ & -U_y^{z-e_y} & \end{pmatrix}, \tag{4.2}$$

where

$$\mathcal{U} = \{U_\mu^z \in U(1),\ \mu = x, y,\ z \in \Omega\} \quad \text{and} \quad m \in \mathbb{R}.$$

Here, z denotes a grid point on the computational domain Ω. The off-diagonal entries of the system matrix are often referred to as gauge configurations which vary according to a specific probability distribution. In our tests, we consider collections of constant gauge variables, namely, $U \equiv$ constant, as well as stochastic distributions. Our labeling of the unknowns and gauge variables is illustrated in Figure A.1, where e_μ denotes the unit vector in the μ-direction; i.e., it describes a shift on the lattice by one lattice site in the μ-direction.

In all tests, we use the same coarsening and sparsity structure of P as for the FE Laplace system (see Figure 4.1) and coarsen the equations until the dimension of the coarsest system is 8×8. We note that the GL operator is Hermitian, and we define m such that $\lambda_{\min} = N^{-2}$, resulting in positive definite yet ill-conditioned system matrices.

4.2.1. Numerical results for the GL with constant U. In section 4.1, we demonstrated the effectiveness of the BAMG-MGE approach for the bilinear FE discretization of Laplace's equation. Here, we test the method for the GL system with various choices of $U \equiv$ constant.

In Table 4.7, we report results for a V^3-cycle setup algorithm applied to the GL system with $U \equiv 1$. Taking $m = 0$, this yields the FD Laplace operator discretized using central finite differences, up to a scaling by h^2.

As the results reported in Table 4.7 show, we obtain a very efficient multigrid solver for this problem with the LSR approach—it reduces the error by an order of magnitude at each iteration, whereas the performance of the LS scheme deteriorates as the problem size is increased.

Table 4.7
Asymptotic convergence rate estimates, ρ, of the MG method obtained by using the LS (LSR) schemes in a V^3-cycle setup with $\eta = 4$ and $k_r = k_e = 8$.

N	32	64	128	256	512
ρ	.080 (.055)	.090 (.056)	.094 (.055)	.089 (.053)	.727 (.052)

As a next test, we apply this same approach for $U \equiv -1$. This operator results from applying a symmetric diagonal scaling to the FD Laplace problem, such that the kernel of the scaled operator alternates between -1 and 1. The results of our experiments are reported in Table 4.8. Again the results suggest that the LSR scheme with a V^3-cycle setup is able to solve this problem efficiently.

Table 4.8
Asymptotic convergence rate estimates, ρ, of the MG method obtained by using the LS (LSR) schemes in a V^3-cycle setup with $\eta = 4$ and $k_r = k_e = 8$.

N	32	64	128	256	512
ρ	.074 (.059)	.080 (.057)	.095 (.055)	.126 (.053)	.998 (.052)

Next, in Table 4.9, we report results for the GL operator with $U \equiv e^{-i\frac{\pi}{7}}$, giving the stencil

$$A(\mathcal{U}) = \begin{pmatrix} & -e^{-i\frac{\pi}{7}} & \\ -e^{i\frac{\pi}{7}} & (4+m) & -e^{-i\frac{\pi}{7}} \\ & -e^{i\frac{\pi}{7}} & \end{pmatrix}.$$

Again, we use a V^3-cycle setup. We note that the resulting system is not equivalent to a diagonal scaling of the FD Laplace operator (see Appendix A for details). Nonetheless, the LSR approach with a V^3-cycle setup yields an efficient solver for this system as well.

Table 4.9

Asymptotic convergence rate estimates, ρ, of the MG method obtained using the LS (LSR) schemes in a V^3-cycle setup with $\eta = 4$ and $k_r = k_e = 8$.

N	32	64	128	256	512
ρ	.058 (.056)	.054 (.054)	.055 (.051)	.051 (.049)	.056 (.048)

4.2.2. Numerical results for the GL with stochastic distributions of the gauge variables. We conclude our experiments with tests for the GL operators $A(\mathcal{U})$ for various realizations of the gauge variables. In interesting cases, they are weakly correlated among neighboring grid points. In general, their distribution depends on a parameter β. The case $\beta = \infty$ yields $U_\mu^z = 1$ for $\mu = x, y$ and all $z \in \Omega$. As we discuss in Appendix A, as the gauge variables become less correlated, the support of the low eigenvectors of $A(\mathcal{U})$ in turn become increasingly local (see Figure A.3). Further, the number of locally supported low modes of $A(\mathcal{U})$ generally increases as the problem size is increased, making it difficult to define an effective MG interpolation operator for the system.

We report asymptotic convergence rates of the stand-alone MG solver and the number of iterations it takes the associated MG preconditioned conjugate gradient method to reduce the initial residual by a factor of 10^8. The results are contained in Table 4.10.

Table 4.10

Asymptotic convergence rate estimates, ρ, of the MG method obtained using the LS scheme (on the left) and LSR scheme (on the right) from V^3- and W-cycle setup algorithms with $\eta = 4$, $k_r = 8$, and $k_e = 16$. The provided integer values denote the number of MG preconditioned conjugate gradient iterations needed to reduce the relative residual by a factor of 10^8.

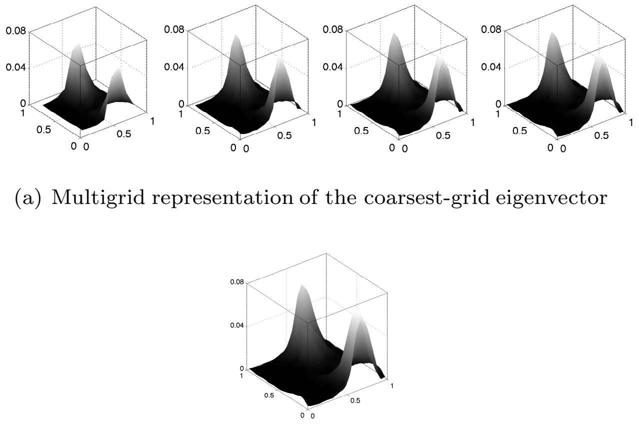
$\beta \setminus N$	32		64		128		256	
1	$.284_W$ 8	$.242_W$ 8	$.416_W$ 9	$.286_W$ 8	$.648_W$ 10	$.478_W$ 9	$.658_{V^3}$ 10	$.493_{V^3}$ 9
5	$.275_W$ 8	$.284_W$ 8	$.264_W$ 8	$.225_W$ 8	$.576_W$ 10	$.389_W$ 10	$.672_{V^3}$ 9	$.469_{V^3}$ 8
10	$.137_W$ 8	$.120_W$ 8	$.225_W$ 7	$.223_W$ 7	$.586_W$ 10	$.349_W$ 9	$.433_{V^3}$ 9	$.423_{V^3}$ 8

An important observation here is that the improved solver performance observed for the LSR approach over the LS one for the GL system with stochastic distributions of the gauge variables is far less pronounced than it was for the FE Laplace system and the GL system with constant gauge variables. Recall that for these tests we apply the adaptive relaxation scheme to all the test vectors before constructing the LSR form of P. We note that even when applying the adaptive relaxation technique to all test vectors, the convergence rates reported for the LSR-based solver are only marginally better than those reported for the LS-based scheme. Overall, the lack of scaling of the stand-alone method when going to larger grid sizes can perhaps be explained by the fact that the coarsest grid is not fine enough to accurately represent all of the locally supported eigenvectors with small eigenvalues.

Due to the stochastic nature of the configurations used in defining the entries of the system matrix, it is not clear that a direct comparison of the method for different problem sizes is relevant. Perhaps a more meaningful observation here is that for the AMG preconditioned conjugate gradient solver the number of iterations is roughly constant for all problem sizes and configurations. This in turn suggests that in the stand-alone multigrid solver only a few error components are not efficiently reduced.

There are several possible ways to remedy this. First, as we did in the tests, we could try to capture as many of the components in our BAMG-MGE setup as possible and then treat the remaining components by recombining successive iterates, for example by using the multigrid solver as a preconditioner for a Krylov subspace method (e.g., the conjugate gradient method). Alternatively, we could treat these locally supported vectors by a block smoother, an idea closer to the mindset of geometric multigrid.

As a final illustration, we demonstrate that even for these more challenging tests the MGE is able to efficiently compute eigenvector approximations of the eigenvectors of the finest-level system. Figure 4.5 contains plots of the modulus of the eigenvector approximations with the smallest eigenvalue computed using a V^2-cycle setup. Again, we observe that the BAMG-MGE cycle efficiently generates an accurate multigrid representation of the eigenvector with the smallest eigenvalue.



(a) Multigrid representation of the coarsest-grid eigenvector

(b) Finest grid

FIG. 4.5. *Comparison of the MGE representation of the eigenvector approximations with the smallest eigenvalue on each level (*a*) with the associated finest-grid eigenvector with the smallest eigenvalue (*b*).*

5. Concluding remarks. In this paper, we developed and tested a bootstrap approach for computing multigrid interpolation operators. As in any efficient multigrid solver, these operators have to be accurate for the lowest eigenvectors of the problem's finest-grid operator. Here, this is achieved by defining interpolation to fit, in an LS sense, a set of test vectors that collectively approximate the algebraically smooth error. We have shown that using LS interpolation with a BAMG cycle, a MGE, and adaptive relaxation leads to an efficient AMG setup algorithm and solver for the scalar test problems considered.

All numerical experiments presented in this paper were for scalar PDEs discretized on structured grids, using full coarsening and interpolation with a fixed sparsity pattern. This allowed us to concentrate on developing and testing techniques for computing the interpolation and the impact of the BAMG setup, the MGE, and the adaptive relaxation on the accuracy of the resulting interpolation operators. Our future research will focus on the use of compatible relaxation as an efficient tool for choosing coarse-grid variables and the use of algebraic distance to define the sparsity structure of the corresponding interpolation operator.

Appendix A. The GL system. The GL operator is a commonly used test problem in AMG algorithm development for lattice formulations of the Dirac equation arising in lattice gauge theories. The lattice Dirac operator describes a discretized system of coupled PDEs. For the sake of definiteness, Wilson's original discretization gives the nearest-neighbor coupling of the unknowns which, for a 2D space and a given $U(1)$ background gauge field, can be written in spin-permuted ordering as

$$D = \begin{pmatrix} A(\mathcal{U}) & B(\mathcal{U}) \\ -B(\mathcal{U})^H & A(\mathcal{U}) \end{pmatrix},$$

where $\mathcal{U}$ denotes a discrete realization of the so-called gauge field. The gauge configuration $\mathcal{U}$ can be understood as a collection of link variables of $U(1)$, i.e., complex numbers with modulus one:

$$\mathcal{U} = \{U_\mu^z \in U(1),\ \mu = x, y,\ z \in \Omega\}.$$

As stated earlier, our labeling of the unknowns and gauge links is illustrated in Figure A.1. Herein, e_μ is the unit vector in the μ-direction; i.e., it describes a shift on the lattice by one lattice site in the μ-direction.

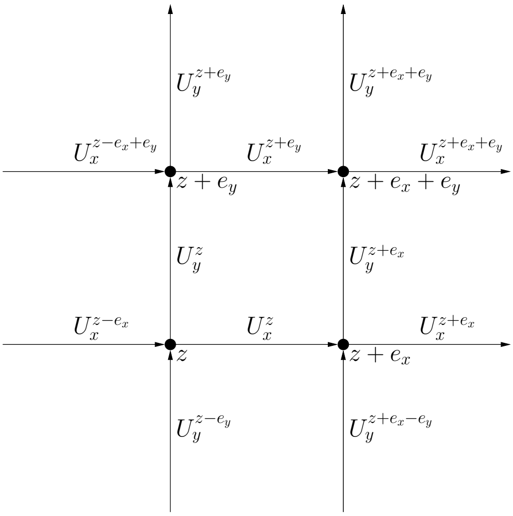


FIG. A.1. *Naming convention on the lattice.*

The diagonal blocks, $A(\mathcal{U})$, are referred to as *GLs*. The action of $A(\mathcal{U})$ on a vector $\psi \in \mathbb{C}^{N^2}$ at a lattice site $z \in \Omega$ reads as

$$\begin{aligned}(A(\mathcal{U})\psi)_z = {} & (4+m)\psi_z - U_x^{z-e_x}\psi_{z-e_x} \\ & - U_y^{z-e_y}\psi_{z-e_y} - \overline{U_x^z}\psi_{z+e_x} - \overline{U_y^z}\psi_{z+e_y},\end{aligned} \tag{A.1}$$

and the action of $B(\mathcal{U})$ on ψ at site $z \in \Omega$ is given by

$$(B(\mathcal{U})\psi)_z = \overline{U_x^z}\psi_{z+e_x} - U_x^{z-e_x}\psi_{z-e_x} + i\left(\overline{U_y^z}\psi_{z+e_y} - U_y^{z-e_y}\psi_{z-e_y}\right).$$

In our numerical experiments, we consider solving the GL system

$$A(\mathcal{U})\psi = \varphi$$

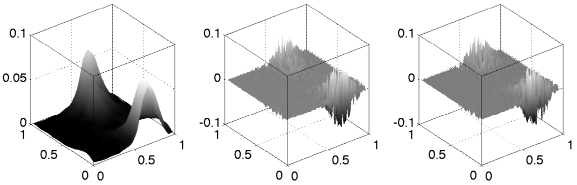


FIG. A.2. *Modulus, real, and imaginary parts of the eigenvector to the smallest eigenvalue for $\beta = 5$ on a 64×64 grid.*

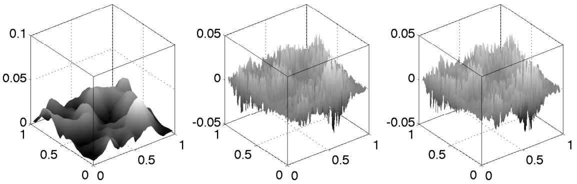


FIG. A.3. *Modulus, real, and imaginary parts of algebraically smooth error after* 50 *GS iterations applied to a random initial guess for $\beta = 5$ on a 64×64 grid, which for our choice of simple pointwise smoother is a linear combination of the low eigenvectors of the fine-grid system.*

for different choices of the shift m and configurations of $\mathcal{U}$. In interesting cases, the off-diagonal entries of $A(\mathcal{U})$ are weakly correlated among neighboring grid points. In general, their distribution depends on a parameter β. The case $\beta = \infty$ yields $U_\mu^z = 1$ for $\mu = x, y$ and all $z \in \Omega$. As $\beta \rightarrow 0$, the gauge variables θ_μ^z in $U_\mu^z := e^{i\theta_\mu^z}$ become less correlated[2] and the support of the low eigenvectors becomes increasingly local.

A.1. Spectral properties of the GL. From (A.1), it follows that the GL matrix is Hermitian. We define the constant, m, so that the resulting matrix A has as its smallest eigenvalue $\lambda_{min} = N^{-2}$, with N denoting the number of grid points in a given direction of the 2D grid. This choice in turn yields positive definite yet ill-conditioned system GL systems.

An important issue to consider when developing solvers for the GL is the local character of its algebraically smooth error. Figure A.2 contains plots of the modulus, real, and imaginary parts of the eigenvector with the smallest eigenvalue of the system matrix in (A.1). In Figure A.3, we provide a plot of the error for $\beta = 5$ on a 64×64 grid computed using 50 GS relaxations. Here, we see two main reasons why standard AMG approaches break down when applied to this problem—the algebraically smooth error is locally supported, and it is not smooth among neighboring grid points in regions where it is nonzero.

To further analyze the properties of the GL operator, we consider the notion of a gauge transformation. A *gauge transformation*

$$\begin{aligned} g : \Omega &\rightarrow U(1), \\ z &\longmapsto g_z \end{aligned}$$

[2] In the reported results, we use gauge variables generated using a code supplied to us by R. Brower from Boston University. The gauge data is available in ASCII format online from *http://www.math.psu.edu/brannick/gaugeLS09*.

of a gauge configuration $\mathcal{U} = \{U_\mu^z\} \subset U(1)$ is defined by

$$U_\mu^z \longmapsto \bar{g}_z U_\mu^z g_{z+e_\mu}.$$

That is, a gauge transformation can be represented as a diagonal matrix:

$$G = \operatorname{diag}(g_z), \quad z \in \Omega.$$

Thus, the action of the gauge transformation on a gauge covariant operator $Z(\mathcal{U})$ is given by

$$Z(\mathcal{U}) \longmapsto G^H Z(\mathcal{U}) G. \tag{A.2}$$

In the case of $\mathcal{U} \subset U(1)$, we obtain the gauge transformation of $\mathcal{U}$ under the gauge transformation $g : z \longmapsto e^{i\psi_z}$ by

$$U_\mu^z = e^{i\theta_\mu^z} \overset{g}{\longmapsto} e^{-i\psi_z} e^{i\theta_\mu^z} e^{i\psi_{z+e_\mu}}. \tag{A.3}$$

Note that the $U(1)$ gauge transformation G in (A.2) fulfills

$$G^H G = I \quad \text{and} \quad \|G_{\cdot,i}\|_2 = 1, \; i = 1, \dots, m;$$

that is, the gauge transformation is a unitary similarity transformation of the matrix $Z(\mathcal{U})$. Hence, if $x_1, \dots, x_m$ are the eigenvectors of $Z(\mathcal{U})$ corresponding to the eigenvalues $\lambda_1, \dots, \lambda_m$, then $G^H x_1, \dots, G^H x_m$ are the eigenvectors of $G^H Z(\mathcal{U}) G$ corresponding to the same eigenvalues. Accordingly, we define an equivalence relation $\sim$ between two operators $Z(\mathcal{U}_1), Z(\mathcal{U}_2)$ with gauge configurations $\mathcal{U}_i$, $i = 1, 2$, as

$$Z(\mathcal{U}_1) \sim Z(\mathcal{U}_2) \iff \text{there exists } G \text{ such that } Z(\mathcal{U}_1) = G^H Z(\mathcal{U}_2) G.$$

Applying this definition, we have that for a given constant configuration ($U_\mu^z \equiv e^{i\theta}$) on an $N \times N$ equidistant lattice, if θ fulfills

$$\theta = \frac{2\pi k}{N} \tag{A.4}$$

for some $k \in \{0, \dots, N-1\}$, then $A(\mathcal{U}) \sim A(\mathcal{U}_0)$, where $\mathcal{U}_0$ denotes the case when $\mathcal{U} \equiv 1$. Thus, for certain distributions of $\mathcal{U}$, $A(\mathcal{U})$ is simply a diagonally scaled Laplacian, whereas, in others, the system is not equivalent. We note that an effective AMG algorithm should solve the Laplace-equivalent systems with a performance similar to its performance for the standard Laplacian, whereas the performance in the nonequivalent case is not predictable. A more detailed study of this system can be found in [14].

REFERENCES

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

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

[3] A. Brandt, *Multiscale scientific computation: Review* 2001, in Multiscale and Multiresolution Methods: Theory and Applications, T. J. Barth, T. F. Chan, and R. Haimes, eds., Springer, Heidelberg, 2001, pp. 1–96.

[4] A. Brandt, S. McCormick, and J. Ruge, *Algebraic Multigrid (AMG) for Automatic Multigrid Solution with Application to Geodetic Computations*, Technical report, Colorado State University, Fort Collins, CO, 1983.

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

[6] J. Brannick, R. Brower, M. Clark, J. Osborn, and C. Rebbi, *Adaptive multigrid algorithm for lattice QCD*, Phys. Rev. Lett., 100 (2008), 041601.

[7] J. Brannick and L. Zikatanov, *Algebraic multigrid methods based on compatible relaxation and energy minimization*, in Domain Decomposition Methods in Science and Engineering XVI, Vol. 55, O. B. Widlund and D. E. Keyes, eds., Springer, Berlin, 2007, pp. 15–26.

[8] J. J. Brannick and R. D. Falgout, *Compatible relaxation and coarsening in algebraic multigrid*, SIAM J. Sci. Comput., 32 (2010), pp. 1393–1416.

[9] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation (αSA)*, SIAM J. Sci. Comput., 25 (2004), pp. 1896–1920.

[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, *A generalized eigensolver based on smoothed aggregation (GES-SA) for initializing smoothed aggregation (SA) multigrid*, Numer. Linear Algebra Appl., 15 (2008), pp. 249–269.

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

[13] K. Kahl, *An Algebraic Multilevel Approach for a Model Problem for Disordered Physical Systems*, Master's thesis, Bergische Universität Wuppertal, Fachbereich Mathematik und Naturwissenschaften, Wuppertal, Germany, 2006.

[14] K. Kahl, *Adaptive Algebraic Multigrid for Lattice QCD Computations*, Ph.D. thesis, Bergische Universität Wuppertal, Fachbereich Mathematik und Naturwissenschaften, Wuppertal, Germany, 2009.

[15] I. Livshits, *One-dimensional algorithm for finding eigenbasis of the Schrödinger operator*, SIAM J. Sci. Comput., 30 (2008), pp. 416–440.

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

[17] T. Manteuffel, S. McCormick, M. Park, and J. Ruge, *Operator-based interpolation for bootstrap algebraic multigrid*, Numer. Linear Algebra Appl., 17 (2010), pp. 519–537.

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

[19] D. Ron, I. Safro, and A. Brandt, *Relaxation Based Coarsening and Multiscale Graph Organization*, preprint, 2010.

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

[21] G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice–Hall, Englewood Cliffs, NJ, 1973.

[22] P. Vassilevski, *Multilevel Block Factorization Preconditioners: Matrix-Based Analysis and Algorithms for Solving Finite Element Equations*, Springer, New York, 2008.

[23] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.