NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2011; **00**:1–18
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nla

Relaxation-corrected Bootstrap Algebraic Multigrid (*r*BAMG) *

M. Brezina[1], C. Ketelsen[2], T. Manteuffel[1], S. McCormick[1], M. Park[3], and J. Ruge[1]

[1] *Department of Applied Mathematics, Campus Box 526, University of Colorado at Boulder, Boulder, CO 80309-0526. email:* {***marian.brezina, tmanteuf, stevem, jruge***}***@colorado.edu***
[2] *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-560, Livermore, CA 94551. email:* ***ketelsen1@llnl.gov***
[3] *School of Mathematical Sciences, University of Nottingham, University Park Nottingham, NG7 2RD. email:* ***min.park@nottingham.ac.uk***

SUMMARY

Bootstrap Algebraic Multigrid (BAMG) is a multigrid-based solver for matrix equations of the form $Ax = b$. Its aim is to automatically determine the interpolation weights used in algebraic multigrid (AMG) by locally fitting a set of test vectors that have been relaxed as solutions to the corresponding homogeneous equation, $Ax = 0$. This paper studies an improved form of BAMG, called *r*BAMG, that involves adding scaled residuals of the test vectors to the least-squares equations.

The basic *r*BAMG scheme was introduced in an earlier paper [1] and analyzed on a simple model problem. The purpose of the current paper is to further develop this algorithm by incorporating several new critical components, and to systematically study its performance on an interesting model problem from quantum chromodynamics (QCD). While the earlier paper introduced a new least-squares principle involving the residuals of the test vectors, a simple extrapolation scheme is developed here to accurately estimate the convergence factors of the evolving AMG solver. Such a capability is essential to effective development of a fast solver, and the approach introduced here is shown numerically to be much more effective than the conventional approach of just observing successive error reduction factors. Another component of the setup process developed here is an adaptive cycling process. This component assesses the effectiveness of the V-cycle constructed in the initial *r*BAMG phase by applying it to the homogeneous equation. When poor convergence is observed, the set of test vectors is enhanced with the resulting error, enabling the subsequent least-squares fit of interpolation to produce an improved V-cycle. A related component is the scaling and recombination Ritz process that targets the so-called weak approximation property in an attempt to reveal the important elements of these evolving error and test vector spaces.

The aim of the numerical study documented here is to provide insight into the various design choices that arise in the development of an *r*BAMG algorithm. With this in mind, the results for QCD focus on the behavior of *r*BAMG in terms of the number of initial test vectors used, the number of relaxation sweeps applied to them, and the size of the target matrices. Copyright © 2011 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: iterative methods, multigrid, algebraic multigrid, adaptive algebraic multigrid

1. INTRODUCTION

Due to its potential to solve an $N \times N$ sparse linear system of equations,

$$Ax = b, \qquad (1)$$

*Portions of this work were performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-JRNL-508373 with only $O(N)$ work, multigrid methods have gained widespread use for solving the large sparse linear systems that arises from the discretization of partial differential equations (PDEs). This popularity is due to the efficiency that results from two complementary processes: *smoothing* and *coarse-grid correction*. The basic idea of the classical geometric multigrid method is that relaxation is inexpensive and efficient at eliminating high-frequency (oscillatory) error, while low-frequency (smooth) error that remains after relaxation can be eliminated by correction from an approximate coarse-grid residual equation, which in turn can be solved by relaxation and correction from yet coarser grids. Unfortunately, many modern problems involve discontinuous coefficients, complex geometries, and unstructured grids, which reduce the effectiveness of geometric multigrid.

In contrast to standard multigrid methods, algebraic multigrid (AMG [2, 3]) methods need not take into account any geometric information about the underlying problem. Instead, AMG relies on a fixed relaxation process, which allows it to focus on extracting information from the given matrix to construct a hierarchy of coarse problems and intergrid transfer operators that address *algebraically smooth errors* (i.e., errors that relaxation cannot efficiently eliminate). In particular, classical AMG is based on the assumption that relaxation cannot efficiently resolve errors that are locally constant. While appropriate use of the characteristics of algebraic smoothness seems essential for obtaining effective solvers, these additional assumptions limit the scope of applicability of AMG methods. What is needed are self-learning algebraic multigrid solvers that automatically determine the full character of algebraically smooth errors.

The most recent research in this direction is concerned primarily with the development of self-learning multigrid algorithms, including the original *adaptive* AMG algorithm introduced in [2], the *bootstrap* AMG approach introduced in [4] and developed further for quantum chromodynamics in [5], an adaptive scheme based on smoothed aggregation (SA) developed in [6] and [7] and developed further for lattice QCD in [8], adaptive AMG schemes developed further in [9], and an adaptive reduction-based AMG algorithm introduced in [10] and [11]. The adaptive approach, as it has usually been implemented, constructs interpolation to initially fit a single test vector, and then tests the resulting multilevel solver on the homogeneous problem, starting from another random initial vector. If observed convergence is not yet acceptable, then the resulting vector is either used to enhance the original test vector or else added to the test-vector set. Interpolation is then adjusted to fit the evolving set of test vectors, and the process then continues until acceptable convergence is observed. BAMG, as it has usually been implemented, instead typically constructs interpolation to fit (in a least-squares sense) several initial test vectors.

Unfortunately, adaptive approaches that begin with a single test vector and introduce additional vectors one at a time can be costly in their initial stages. This can happen because an effective coarse-grid solver must be developed before returning to the fine grid if the quality of interpolation is to be safely assessed. Otherwise, it would be difficult to determine whether slow convergence is due to a poor interpolation operator or a poor coarse-level solver. To address this concern, we develop a version here of adaptive AMG that begins with several test vectors that are chosen randomly and subjected to relaxation (for solving the homogeneous equation), and are then used in a modified least-squares process of fitting interpolation. While this version fits into the adaptive methodology, we refer instead to it as a BAMG scheme because using multiple initial vectors is more in the spirit of the bootstrap methodology.

The aim of the least-squares process is to produce a V-cycle for solving (1) that converges quickly in a sense that we describe below. Our approach is to use the initially constructed interpolation operator to create a V-cycle and, in the subsequent adaptive phase, test this current solver applied to a random initial guess for the homogeneous equation, $Ax = 0$. For focus and simplicity of analysis, we define this adaptive phase in a non-recursive way in that we only test the effectiveness of the current solver on the finest level. As we said, the usual adaptive AMG approach is to test solver effectiveness recursively on all levels, and not return to finer levels until adequate performance is observed on the coarse levels. A related recursive approach base on coarse-level eigenvector approximations was developed for BAMG in [5] that seems to dramatically reduce the amount of necessary coarse-level cycling. However, we choose a simple non-recursive form of the adaptive phase because it facilitates a systematic focus on the other components of the AMG processes.

DOI: 10.1002/nla

The focus of this paper is on developing more efficient self-learning algorithms that are able to fit several smooth vectors by least-squares approximation. Because of this focus, we obviate the problem of choosing a good coarse grid by considering cases where this is known in advance. In general, selection of the coarse grid could be obtained by a compatible relaxation process; see [12], [13], [5], and[14]. The study of *r*BAMG here is an attempt to systematically analyze the behavior of the algorithm in terms relative to several parameters. The focus is on the number of test vectors, the number of relaxation sweeps applied to them, and the dimension of the matrix to which the scheme is applied. A large number of other parameters and options could also be considered, including different cycling strategies, other coarsening strategies (e. g., computing several eigenvector approximations on coarse levels), different numbers of relaxation sweeps on coarse levels, different possible strategies for combining test vectors and error components produced by the current cycles, and so on. Studying all of these options and parameters would not be feasible here. Instead, reasonable choices are made based on some sample studies (that, in the interest of space, we choose not to document here), with the hope that the *r*BAMG algorithm studied here is fairly effective and robust. Our numerical analysis is thus able to concentrate on how this scheme behaves numerically in the face of increasing the numbers of test vectors and relaxation sweeps performed on them. A similar test for Poisson's equation using a fixed problem size was described in [5] (including extensive comparisons of *r*BAMG and BAMG for Poisson and gauge Laplacian problems). Our principal focus is, however, to observe the optimal choice for the number of test vectors and relaxation sweeps as a function of problem size, which we do here in the context of the shifted gauge Laplacian.

2. CLASSICAL AMG

Assume for the remainder of this paper that $A \in \Re^{n\times n}$ is symmetric and positive definite. This section summarizes how classical AMG applied to $Ax = b$ determines the weights in the interpolation operator, which we denote by P. Since the focus is on the process of determining interpolation weights, we assume that the fine-level points have already been partitioned into the C-points that are identified with the coarse grid and its F-point complement set. This partition into C and F points gives rise to the AMG form of interpolation described as follows: the ith entry of Pe is given by

$$(Pe)_i = \begin{cases} e_i & \text{if } i \in C, \\ \sum\limits_{j\in C_i} w_{ij} e_j & \text{if } i \in F. \end{cases} \tag{2}$$

Here, C_i is the subset of C of points that are used to interpolate to point $i \notin C$. Our task now is to describe in abstract terms how the interpolation weights, w_{ij}, are determined. Therefore, for the remainder of this section, we consider a given fixed $i \in F$.

Interpolation only needs to approximate error that is not easily attenuated by relaxation. This observation gives rise to the first AMG premise: relaxation produces error, e, that is algebraically smooth in the sense that it exhibits a relatively small residual. Therefore, we can assume that $(Ae)_i \approx 0$, that is, that

$$a_{ii} e_i \approx -\sum_{j\neq i} a_{ij} e_j.$$

Consider now the splitting of this sum into its component sums over C_i, the coarse interpolatory set, and C_i^c, the complement of C_i in the *neighborhood* of i, by which we mean the set of points that are connected to i in A (i. e., all points ℓ such that $a_{i\ell} \neq 0$). Note that C_i^c consists of connected F points and other connected C points that are not in C_i. With this splitting, we obtain

$$a_{ii} e_i \approx -\sum_{j\in C_i} a_{ij} e_j - \sum_{k\in C_i^c} a_{ik} e_k. \tag{3}$$

Note that, if the second component sum happens to vanish in this approximation (e. g., $a_{ik} = 0$ for $k \in C_i^c$), then we would immediately have a formula that expresses the value of any algebraically smooth error at point i by its value at points of C_i. This “operator interpolation” formula would then yield appropriate weights for P given by $w_{ij} = -a_{ij}/a_{ii}$. This observation suggests, for the general case $\sum_{k\in C_i^c} a_{ik}e_k \neq 0$, that we need to “collapse” the unwanted connections (a_{ik} for $k \in C_i^c$) to C_i. Thus, we need to replace the e_k in the second sum on the right side of (3) with sums that involve only e_j for $j \in C_i$.

To replace each e_k, $k \in C_i^c$, with a linear combination of the e_j, $j \in C_i$, we need to make a further assumption about the nature of smooth error. Since the historical target for AMG is partial differential equations of elliptic type, the classical AMG premise is that smooth error is locally almost constant. This second AMG premise means that we can assume that each e_k is any convex linear combination of the e_j, $j \in C_i$, that preserves constants. AMG is based on the particular linear combination where the coefficients are proportional to the connections from point k to each point j, that is, it is based on the approximation

$$e_k \approx \sum_{j\in C_i} \frac{a_{kj}}{\sum\limits_{\ell\in C_i} a_{k\ell}} e_j. \tag{4}$$

Substituting this expression into (3) and dividing the result by a_{ii} yields interpolation weights given by

$$w_{ij} = \frac{1}{a_{ii}}\left(-a_{ij} - \sum_{k\in C_i^c}\left(a_{ik}\frac{a_{kj}}{\sum\limits_{\ell\in C_i} a_{k\ell}}\right)\right). \tag{5}$$

This process of collapsing the unwanted connections in the operator interpolation formula expressed by (3) can be viewed as using a crude but properly scaled *truncated* interpolation formula, expressed by (4), to interpolate from C_i to e_k. This indirect process has the effect of collapsing the unwanted connections, and it leads to the direct formula for interpolation weights defined in (5).

3. BAMG AND *r*BAMG METHODS

Suppose now that we are given a set of test vectors, $e^{(l)}$, $l = 1, 2, \dots, q$, that result from several fine-level relaxation sweeps on the homogeneous equation, $Ae = 0$, starting from q distinct random approximations. (We assume that the initial random vectors are of unit length in the Euclidean norm to avoid significant scale disparity in the least-squares processes that follow.) Assume that the coarse-grid interpolatory set, C_i, has already been determined for each F-point i. Assume also, for simplicity, that the vectors are locally rich in that they are locally independent and numerous enough to ensure that all of the least-squares problems we introduce are uniquely solvable. (This is the case for the tests documented here. See [1] for the general case.)

3.1. BAMG

The general form of the interpolation operator, P, for AMG is given in (2). As described in the previous section, the choice of weights, w_{ij}, is typically dictated by two basic premises: relaxation produces small residuals and relaxed errors are locally almost constant. The first premise is very general: many matrix equations can be treated by relaxation schemes that produce small (possibly weighted or preconditioned) residuals in a sense that leads to usable local operator interpolation formulas. However, many circumstances arise where local errors are not approximately constant in any local sense, so the second premise seriously restricts the applicability of AMG methods. The basic idea behind adaptive AMG and BAMG is to glean the local character of algebraically smooth errors from the set of test vectors. This leads to a determination of the interpolation weights by direct least-squares fit of the target vectors. Thus, for each $i \in F$, we compute

$$\text{(BAMG)} \qquad \{w_{ij} : j \in C_i\} = \arg\min_{w_{ij}} \sum_{l=1}^{q} (e_i^{(l)} - \sum_{j\in C_i} w_{ij} e_j^{(l)})^2. \tag{6}$$

BAMG takes a direct approach to determining the weights of interpolation. An indirect approach that is more in the spirit of classical AMG is what leads us to *r*BAMG.

3.2. rBAMG

The least-squares problem for BAMG given by (6) can be modified by the addition of the local scaled residual as follows:

$$(r\text{BAMG}) \qquad \{w_{ij} : j \in C_i\} = \arg\min_{w_{ij}} \sum_{l=1}^{q} (e_i^{(l)} - \sum_{j \in C_i} w_{ij} e_j^{(l)} - \frac{r_i^{(l)}}{a_{ii}})^2, \tag{7}$$

where $r_i^{(l)} = \sum_{j=1}^{n} a_{ij} e_i^{(l)}$. This change to the fitting process yields a new scheme, which we call *r*BAMG, that is equivalent to the indirect BAMG (*i*BAMG) approach introduced in [1] for the case that unwanted connections are collapsed to all of C_i and the target vectors are rich enough locally to guarantee a unique fit. This change should, therefore, improve the direct approach insofar as numerical tests show the superiority of *i*BAMG in [1]. Thus, we can expect this improved approach to offer better performance for a given number of target vectors and relaxation steps applied to them.

Note that this modification to the direct scheme is equivalent to relaxing on the homogeneous equation at point i for each target vector and then applying the standard BAMG minimization approach. As such, *r*BAMG is related in spirit to the adaptive relaxation scheme described by Brandt in [15] (and suggested in [12]) that applies relaxation selectively to points exhibiting especially large residuals. A subtle but important difference here is that, to determine interpolation at point i, the residuals are effectively set to zero just at i. This local process is hopefully more effective than attempting to drive all residuals to near zero globally. (What appears to be a very efficient compromise between *r*BAMG and adaptive relaxation was introduced in [5] that incorporates in (6) only a few of the largest residuals for only the most algebraically smooth target vector.)

3.3. iBAMG Theory

In this subsection, we develop a basic theory for BAMG and *i*BAMG convergence in an AMG reduction-based framework (AMGr). The idea behind AMGr is to first assume that the F-point equations are solved exactly in relaxation. (See the comment at the end of this section on the more practical case that the F-point equations are only approximately solved.) We show here that BAMG and *i*BAMG provide good exact coarse-grid correction in the AMGr context, with conditions that confirm when *i*BAMG is superior.

We assume that the target matrix is symmetric and positive definite, has only 1's on the diagonal (otherwise, we simply diagonally scale it), and is represented in F-C block form as follows:

$$A = \begin{bmatrix} I - X & -Y \\ -Y^T & I - Z \end{bmatrix}.$$

Letting BAMG interpolation be expressed as

$$P_B = \begin{bmatrix} B \\ I \end{bmatrix}, \tag{8}$$

then a variant of *i*BAMG interpolation (that collapses F-F connections to C_j as opposed to C_i, that is, the e_k in the second sum on the right side of (2.2) is replaced by sums that involve only e_l for $l \in C_j$) is given by

$$P_I = \begin{bmatrix} XB + Y \\ I \end{bmatrix}. \tag{9}$$

'Ideal' interpolation is given by

$$\widehat{P} = \begin{bmatrix} (I - X)^{-1} Y \\ I \end{bmatrix}. \tag{10}$$

DOI: 10.1002/nla

(We refer to $\widehat{P}$ here as ideal because exact F-point relaxation produces error that is in the range of $\widehat{P}$, which means that an exact coarse-grid correction based on $\widehat{P}$ produces an exact solver.) First notice that an exact F-point relaxation step is monotonically nonincreasing in energy because it is just an energy-orthogonal projection of the error onto the range of $\widehat{P}$. Two-grid convergence based on P_B or P_I can, thus, be established simply by showing that coarse-grid correction reduces the error in the range of $\widehat{P}$. That is, with subscript A denoting the energy norm ($||x||_A = (x^T Ax)^{1/2}$), we can assume that the initial error satisfies

$$\begin{aligned} \|\widehat{P}e\|_A^2 =& < \begin{bmatrix} (I-X)^{-1}Y \\ I \end{bmatrix} e, A \begin{bmatrix} (I-X)^{-1}Y \\ I \end{bmatrix} e > \\ =& < \begin{bmatrix} (I-X)^{-1}Y \\ I \end{bmatrix} e, \begin{bmatrix} 0 \\ I-Z-Y^T(I-X)^{-1}Y \end{bmatrix} e > \\ =& < e, Se >, \end{aligned}$$

where we have denoted the Schur complement by $S = I - Z - Y^T(I-X)^{-1}Y$.

Now, using P_B to do an exact coarse-grid correction on this error (that is, applying $T_B = I - P_B(P_B^T AP_B)^{-1}P_B^T A$), we compute the squared-energy convergence factor by

$$\begin{aligned} & \frac{\|T_B\widehat{P}e\|_A^2}{\|\widehat{P}e\|_A^2} \\ =& \frac{< AT_B\widehat{P}e, T_B\widehat{P}e >}{< A\widehat{P}e, \widehat{P}e >} \\ =& \frac{< A\widehat{P}e, \widehat{P}e > - < AP_B(P_B^T AP_B)^{-1}P_B^T A\widehat{P}e, \widehat{P}e >}{< A\widehat{P}e, \widehat{P}e >} \\ =& 1 - \frac{< (P_B^T AP_B)^{-1}Se, Se >}{< e, Se >}. \end{aligned}$$

Then, the worst-case squared-energy convergence factor is the smallest $\epsilon_B \geq 0$ for which

$$(P_B^T AP_B)^{-1} \geq (1-\epsilon_B)S^{-1}.$$

(By the relation $E \geq F$ (or $F \leq E$) between symmetric matrices E and F, we mean that $< x, Ex > \geq < x, Fx >$ for all x.)

Note that $\epsilon_B \leq 1$ because $(P_B^T AP_B)^{-1}$ is SPD and $\epsilon_B \geq 0$ because $S^{-1} \geq (P_B^T AP_B)^{-1}$. This last inequality follows because it is equivlent to $P_B^T AP_B \geq S$, which follows from the observation that

$$P_B^T AP_B = S + (P_B - (I-X)^{-1}Y)^T(I-X)(P_B - (I-X)^{-1}Y) \geq S.$$

We can now rewrite this bound as

$$P_B^T AP_B \leq \frac{1}{1-\epsilon_B}S. \tag{11}$$

The key now is to realize that the best P_B, in terms of speed of convergence, is the ideal one, so we write its F block as a modification of the F block of $\widehat{P}$:

$$B = (I-X)^{-1}Y + E_B. \tag{12}$$

Substituting (12) into (11), we obtain

$$E_B^T(I-X)E_B \leq \frac{\epsilon_B}{1-\epsilon_B}S. \tag{13}$$

We can just assume that this holds for some $\epsilon_B < 1$ if all we want to do is compare *i*BAMG to BAMG. Following this line of thinking, we get a similar expression for P_I, of course, except with

$B_I = XB + Y$ replacing B:

$$E_I^T(I - X)E_I \leq \frac{\epsilon_I}{1 - \epsilon_I} S,$$

where ϵ_I is the squared-energy convergence factor for *i*BAMG. It is interesting that E_B and E_I have a very simple relationship:

$$E_I = XE_B. \tag{14}$$

So, for *i*BAMG, a little algebra shows that we want the smallest ϵ_I for which

$$E_B^T X(I - X)XE_B \leq \frac{\epsilon_I}{1 - \epsilon_I} S. \tag{15}$$

Now, the left sides of (13) and (15) are related as follows:

$$E_B^T X(I - X)XE_B \leq \delta^2 E_B^T(I - X)E_B, \tag{16}$$

where δ bounds X in the sense that $-\delta \leq X \leq \delta I$. Note that δ can be interpreted as the fraction of strength represented in C_i^c relative to C_i, so we can assume that δ is bounded by a constant that is less than 1 (e.g., $\delta \leq \frac{1}{2}$ for the 9-point Laplacian and standard coarsening). Now, bounds (13) and (16) imply that

$$E_B^T X(I - X)XE_B \leq \frac{\delta^2 \epsilon_B}{1 - \epsilon_B} S.$$

Since ϵ_I is the smallest quantity satisfying (13), we must have that

$$\frac{\epsilon_I}{1 - \epsilon_I} \leq \frac{\delta^2 \epsilon_B}{1 - \epsilon_B} S.$$

One can then conclude (after a little more algebra) that

$$\epsilon_I \leq (\frac{\delta^2}{1 - \epsilon_B + \delta^2 \epsilon_B})\epsilon_B.$$

Note that the factor in front of ϵ_B here is always less than 1 by our assumption on δ. Thus, *i*BAMG always produces a better convergence factor than BAMG. More importantly, when BAMG converges fairly well, that is, when ϵ_B is somewhat small, then this factor is approximately bounded by $\delta^2 \epsilon_B$. Thus, the (*unsquared*) energy convergence factor for *i*BAMG in this case is roughly δ times that of BAMG.

We want to emphasize that *i*BAMG, in contrast to classical BAMG, can provide reasonable interpolation even when the test vectors are poor. For example, consider the case of a five-point stencil and red-black coarsening. Since no F points have immediate F-point neighbors, then no test vectors are needed because there is no work for *i*BAMG to do. Said differently, this case is characterized by $X = 0$, and *i*BAMG therefore yields the ideal interpolation operator:

$$P_I = \widehat{P} = \begin{bmatrix} Y \\ I \end{bmatrix}.$$

This case is, of course, very special, but it emphasizes the point that *i*BAMG does not necessarily need especially smooth test vectors, particularly when the F points are very strongly connected to the C points.

We should also emphasize that the *i*BAMG scheme studied here is not what we typically use in practice. Here we allowed the stencil to grow to include $C_j - C_i$, which is typically nonempty. Our usual approach is to restrict interpolation to C_i. Thus, the *i*BAMG scheme studied here is generally more expensive than its BAMG counterpart. (Note also that we cannot generally equate *i*BAMG to BAMG when the test vectors have zero F-point residuals.) Nevertheless, this simplified analysis gives insight into the superiority of our more practical *i*BAMG approach.

The theory here assumed exact F-point relaxation, which presumably requires exact inversion of $I - X$. It is more practical to assume that the F-point equations are only approximately solved.

To study this case, we can write the current error in the energy-orthogonal decomposition form $\widehat{P}e + Rz$, where $R = \begin{pmatrix} I \\ 0 \end{pmatrix}$. Now, relaxation on the F points is presumably local, so its effect on the error is to change only z. If Rz is large compared to $\widehat{P}e$ in energy, then relaxation alone can effectively reduce the error because $I - X$ is well conditioned (by assumption on δ). We can therefore assume that we apply a fixed but sufficient number of F-point relaxations to ensure that Rz is smaller in energy relative to $\widehat{P}e$ than any desired factor, η. We can thus conclude that the effect of a coarse-grid correction based on P_B is dominated by $\widehat{P}e$:

$$\|T_B(Pe + Rz)\|_A \leq \|T_B\widehat{P}e\|_A + \|T_BRz\|_A \leq (1+\eta)\|T_B\widehat{P}e\|_A$$

and, similarly,

$$\|T_B(\widehat{P}e + Rz)\|_A \geq (1-\eta)\|T_B\widehat{P}e\|_A.$$

Using this expression, together with an analogous result for the coarse-grid correction based on P_B, allows us to modify the theoretical bounds presented above by an arbitrarily small quantity, η. This, in turn, allows us to apply theory to the practical case that the F-point equations are only treated by simple relaxation, provided we assume a sufficient number of relaxation sweeps.

4. THE ADAPTIVE AMG ALGORITHM

The importance of the two complementary multigrid principles of smoothing and coarse-level correction cannot be stressed enough. The development of an efficient multigrid method depends on a proper balance between these two processes. The main idea behind the adaptive approach is to improve interpolation based on slow-to-converge components. Unlike standard adaptive AMG that typically begins with one test vector, we choose here to begin by relaxing ν times on a number of random test vectors as initial guesses for solving the homogeneous problem, $Ax = 0$. We then compute an interpolation operator, P, based on a least-squares fit of the target vectors. The coarse-grid operator is then formed by the usual Galerkin approach, P^TAP. On the coarse grid, the pre-images of the fine-grid test vectors under interpolation P (which, for AMG, are just their restrictions to the C points) are used as an initial set of test vectors. These q vectors are in turn relaxed ν times (in terms of the homogeneous Galerkin coarse-grid problem) and used in a similar least-squares process to define interpolation to a yet coarser grid. This process continues to the coarsest level (determined in advance to contain just a few points), where no further processes or test vectors are needed in the setup because coarsening is not needed there. This completes the initial setup phase. Note here that we fix the number of test-vector iterations over all levels.

Once an initial AMG hierarchy has been computed, we test the current method by running several $V(1,1)$ cycles. Since these tests can contribute significantly to overall cost, it is best to keep their number as small as possible. To achieve this, it is necessary to develop a practical estimator that would reliably gauge the quality of the current solver after a very few iterations of the solver. Our approach is to use a simple extrapolation scheme that is based on just a few observed cycles of the current solver. Our aim, then, is to develop a very simple model of convergence that involves just a few parameters that can be determined from very few observed convergence factors, typically three or four in our case.

In the next subsection, we introduce just such a convergence model that is used to estimate the asymptotic convergence factors of the current solver, and we describe there how these estimates are used to determine the progress of the adaptive process. This model also includes a formula for estimating the cost of the overall solution process. The solver costs are computed in a separate test phase that applies V(1,1) cycles of the final solver to a random initial guess for $Ax = 0$. The solver costs are computed based on using the observed asymptotic convergence factor to determine how many of these cycles would be needed to reduce the error for a general linear solve by ten orders of magnitude. These cost estimates are used in the setup process to monitor current and projected future costs, which allows us to determine whether expected improvements in the convergence factors we estimate are worth continuing the adaptive cycles.

DOI: 10.1002/nla

The overall flow of our approach is shown in Algorithm 4.1. Note that it assumes two input parameters, ρ_{good} and ρ_{bad}, that guide our judgement as to whether the estimated convergence factor, ρ_{est}, of the current solver is acceptable ($\rho_{est} \leq \rho_{good}$), unacceptable ($\rho_{bad} < \rho_{est}$), or indeterminate ($\rho_{good} < \rho_{est} < \rho_{bad}$). In the first case, we terminate the process; in the second, we continue; and, in the third, we use the work estimate to decide whether to terminate or continue. This algorithm also assumes three additional input cycling parameters, α, ν_1, and ν_2, that determine the form of the current solver. Specifically, we define the current solver to consist of indivisible α $V(\nu_1, \nu_2)$ cycles, meaning that we are testing the overall convergence factor for α applications of a V-cycle that use ν_1 relaxation sweeps on the descent through coarser levels and ν_2 relaxations on the ascent back to the fine grid. Our experience with several numerical tests of other options on the problems we study below suggest that $\alpha = 4$ and $\nu_1 = \nu_2 = 1$ are reasonable choices to make in terms of overall efficiency of the setup and solver process. These choices are what we use in all of the experiments documented in this paper.

Another important choice we make is to apply a Ritz process to the set that includes the initial relaxed test vectors and the evolving error components produced by the current solver. The purpose of the Ritz process is to sort out the various levels of smoothness in the subspace of target vectors. This sorting is needed because fitting interpolation must pay more attention to smooth vectors than it does to oscillatory ones. This principle is articulated in the so-called weak approximation property that guides us in the determination of the proper weights used in the least-squares process. See [7] for a discussion of this principle and how it is used in the context of adaptive smoothed aggregation.

Algorithm 4.1 Non-recursive adaptive AMG algorithm

```
for $j = 0$ to *maxAdapt* do
  for $k = 1$ to *coarsest* - $1$ do
    if $k == 1$ then
      if $j == 0$ then
        Pick the set of $q$ random vectors, $\{x^{(1)}_{(1)}, ..., x^{(1)}_{(q)}\}$
      end if
    else
      Let $x^{(k)}_{(i)} = (x^{(k-1)}_{(i)})_c$, $i = 1, ..., q + j$.
    end if
    if $j > 0$ then
      RITZ($A^{(k)}, \{x^{(k)}_{(1)}, ..., x^{(k)}_{(q+j)}\}$)
    end if
    if $j == 0$ OR ($j > 0$ AND $k > 1$) then
      Relax on $A^{(k)} x^{(k)}_{(i)} = 0$ $\nu$ times, $i = 1, ..., q + j$.
    end if
    Compute interpolation, $I^k_{k+1}$.
    Compute the coarse-grid operator, $A^{(k+1)} = (I^k_{k+1})^T A^{(k)} I^k_{k+1}$.
  end for
  Pick random initial $x^{(1)}$.
  Apply $\alpha$ V($\nu_1$,$\nu_2$) cycles to the homogeneous fine-grid problem, $A^{(1)} x^{(1)} = 0$.
  Estimate the convergence factor, $\rho_{est}$.
  Compute $ncycle$ such that $\rho_{est}^{ncycle} < tol$.
  Compute the total cost, $W^{total}_j = W^{setup}_j + W^{percycle}_j \times ncycle$.
  if $\rho_{est} \leq \rho_{good}$ then
    stop.
  else if $\rho_{est} > \rho_{bad}$ then
    $x^{(1)}_{q+j+1} \longleftarrow x^{(1)}$.
    continue.
  else
    if $W^{total}_j > W^{total}_{j-1}$ then
      stop
    else
      $x^{(1)}_{q+j+1} \longleftarrow x^{(1)}$.
      continue.
    end if
  end if
end for
```

4.1. Convergence Estimation

Access to a simple and efficient stopping criterion is an essential ingredient of our adaptive methodology, the core component of which is a reliable estimate of the quality of the current solver. We could obtain such an estimate simply by applying the current method enough times to the homogeneous equation to ensure that a reliable measure of the worst-case convergence factor can be observed. Unfortunately, in practice, this naive approach usually requires so many cycles that it has a deleterious effect on overall complexity of the setup process. Our approach instead is to use a simple extrapolation scheme that is based on just a few observed cycles of the current solver. Our aim, then, is to develop a very simple model of convergence of the solver that involves just a few parameters that can be determined by very few observed convergence factors, typically three or four in our case.

Specifically, the convergence model we develop here is based on the assumptions that each error is dominated by two orthonormal components, e_1 and e_2, and that these are eigenvectors of the cycle's error propagation matrix with eigenvalues β_1 and β_2, respectively ($\beta_1 > \beta_2$). Suppose now that the error after k cycles, $e^{(k)}$, is a linear combination of these two orthogonal components with coefficients α_1 and α_2, respectively. Convergence of the method is then dominated by β_1. Since there are four unknowns, α_1, α_2, β_1, and β_2, we need at least four consecutive errors to determine them:

$$\begin{aligned} e^{(k)} &= \alpha_1 e_1 + \alpha_2 e_2, \\ e^{(k+1)} &= \alpha_1 \beta_1 e_1 + \alpha_2 \beta_2 e_2, \\ e^{(k+2)} &= \alpha_1 \beta_1^2 e_1 + \alpha_2 \beta_2^2 e_2, \\ e^{(k+3)} &= \alpha_1 \beta_1^3 e_1 + \alpha_2 \beta_2^3 e_2. \end{aligned} \tag{17}$$

We first compute the squared Euclidean norm of these four consecutive errors. Using orthogonality of the two components, we obtain

$$\begin{aligned} ||e^{(k)}||^2 &= \alpha_1^2 + \alpha_2^2, \\ ||e^{(k+1)}||^2 &= \alpha_1^2\beta_1^2 + \alpha_2\beta_2^2, \\ ||e^{(k+2)}||^2 &= \alpha_1^2\beta_1^4 + \alpha_2\beta_2^4, \\ ||e^{(k+3)}||^2 &= \alpha_1^2\beta_1^6 + \alpha_2\beta_2^6. \end{aligned} \tag{18}$$

For the sake of convenience, define $c_i = ||e^{(k+i)}||^2$, $i = 0, ..., 3$, $a_j = \alpha_j^2$, and $b_j = \beta_j^2$, j = 1,2. Then

$$\begin{aligned} c_0 &= a_1 + a_2, \\ c_1 &= a_1 b_1 + a_2 b_2, \\ c_2 &= a_1 b_1^2 + a_2 b_2^2, \\ c_3 &= a_1 b_1^3 + a_2 b_2^3. \end{aligned} \tag{19}$$

Since β_1 (or, equivalently, b_1) is the quantity of interest in the estimation and prediction model, we eliminate unknowns a_1 and a_2 in turn. Multiplying the first equation by -b_1 and adding the result to the second equation, and similarly for the third and fourth equation, we obtain

$$\begin{aligned} c_1 - c_0 b_1 &= a_2(b_2 - b_1), \\ c_2 - c_0 b_1^2 &= a_2(b_2^2 - b_1^2), \\ c_3 - c_0 b_1^3 &= a_2(b_2^3 - b_1^3). \end{aligned} \tag{20}$$

Similarly, we eliminate a_2, arriving at

$$\begin{aligned} c_0 b_1 b_2 - c_1(b_1 + b_2) &= -c_2, \\ c_1 b_1 b_2 - c_2(b_1 + b_2) &= -c_3, \end{aligned} \tag{21}$$

Thus, b_1 is the larger root of the following equation:

$$x^2 - \gamma x + \delta = 0, \tag{22}$$

where δ and γ solve

$$\begin{bmatrix} c_0 & -c_1 \\ c_1 & -c_2 \end{bmatrix} \begin{bmatrix} \delta \\ \gamma \end{bmatrix} = - \begin{bmatrix} c_2 \\ c_3 \end{bmatrix}.$$

4.2. Numerical Illustrations of the Convergence Estimation

In this subsection, we test the performance of our convergence estimation model. We do this by first applying just the *r*BAMG setup process using two different extreme values for the number of test vectors, q, and number of relaxation sweeps, ν, applied to them, namely, $q = \nu = 1$ and $q = \nu = 10$. Our target problem here is the 5-point finite-difference discretization of the gauge Laplacian (c.f., [16]) in two dimensions on a uniform, doubly periodic $N \times N$ grid with red-black coarse grids, shifted by subtracting a scalar multiple of the identity matrix to make the problem more ill-conditioned. Gauge Laplacian matrices are usually scaled so that their entries are of order 1, but we typically rescale them by multiplying by $\frac{4}{h^2} = 4N^2$ so that they have $\frac{4}{h^2}$ on the diagonal and correspond more closely to a standard Laplacian (the so-called 'cold' case). With this scaling, the shift is chosen so that the smallest eigenvalue of the matrix is approximately $\frac{1}{N^2}$, yielding a matrix condition number that is $O(N^4)$.

The gauge fields were generated using the heat-bath algorithm with heat parameter $\beta = 2$. This algorithm is a type of Markov Chain Monte Carlo method that generates a sequence of gauge configurations converging to a desired probability distribution. See [17]. The shift is computed by an implicitly restarted Arnoldi iteration in ARPACK for computing the smallest matrix eigenvalue.

We first estimate the asymptotic convergence factors of the resulting $V(1,1)$ cycles for $N = 64$ by running 100 cycles applied to the homogeneous problem and observing the convergence factor for each cycle. See Figures 1a and 1b. Note that these figures also clearly illustrate that it takes many iterations to obtain a reliable estimate of the worst-case convergence factor by observing individual factors alone. Tables I and II compare the asymptotic convergence factors observed in Figure 1a and 1b, respectively, to the estimations obtained from our extrapolation process. Parameter k in Tables I-IV refer to where we estimated the convergence factors: k = 0 refers to the first four test V(1,1) cycles, k = 1 to cycles 2 through 5, and k = 2 to cycles 3 through 6. Note that, as Table I shows, our predictions based on just a few cycles are quite accurate even for the somewhat slowly converging process shown in Figure 1a. This quality of the estimation degrades in the face of poor solver convergence, as Table II shows. However, the ability of our estimation to signal the extremes in terms of good and bad convergence is the key here and the approach is very effective in this regard.

Asymptotic Convergence Factor	k = 0	k = 1	k = 2
0.77	0.76	0.78	0.78

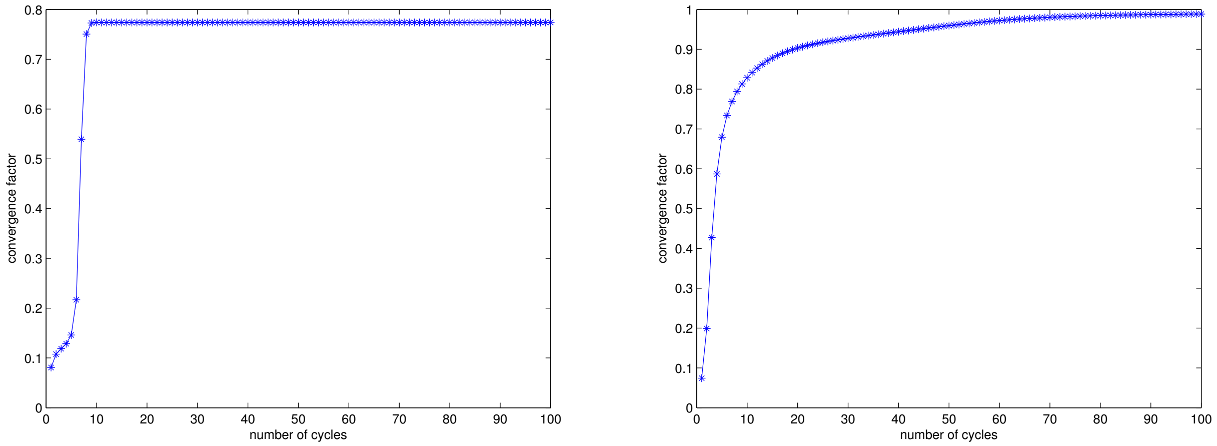
Table I. Asymptotic convergence factor for V(1,1) cycles and its approximations using *r*BAMG with q= ν=10.

Asymptotic Convergence Factor	k = 0	k = 1	k = 2
0.99	0.77	0.84	0.88

Table II. Asymptotic convergence factor for V(1,1) cycles and its approximations using *r*BAMG with q= ν=1.

To carry out these tests further, we proceed in the same way but incorporate a single adaptive cycle in the setup after the least-squares fit of interpolation. We, thus, apply 4 $V(1,1)$ cycles to a random initial guess for $Ax = 0$ and combine the result with the initial test vectors in a Ritz process.

DOI: 10.1002/nla



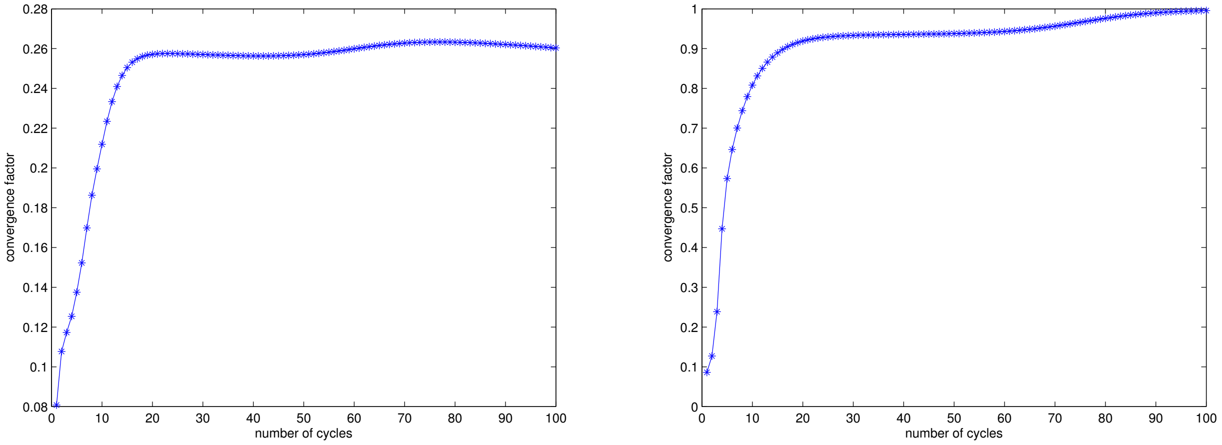
(a) $q = \nu = 10$

(b) $q = \nu = 1$

Figure 1. Convergence factors for V(1,1) cycles using *r*BAMG.

We observe similar behavior of the ability of the extrapolation process to estimate the convergence factor fairly quickly, especially in the important case of reasonably good convergence of the current solver. See Tables III and IV.

Our experience here, and in other tests, suggests that our extrapolation process reduces the number of cycles needed to obtain a reliable estimate by typically a factor of two or more.



(a) $q = \nu = 10$

(b) $q = \nu = 1$

Figure 2. Convergence factors for V(1,1) cycles using *r*BAMG and incorporating a single adaptive cycle.

Asymptotic Convergence Factor	k = 0	k = 1	k = 2
0.26	0.22	0.22	0.26

Table III. Asymptotic convergence factor for V(1,1) cycles and its approximations using *r*BAMG with q= ν=10 and incorporating a single adaptive cycle.

Asymptotic Convergence Factor	k = 0	k = 1	k = 2
0.99	0.76	0.84	0.90

Table IV. Asymptotic convergence factor for V(1,1) cycles and its approximations using *r*BAMG with q= ν=1and incorporating a single adaptive cycle.

5. NUMERICAL EXPERIMENTS: SHIFTED GAUGE LAPLACIAN

Our numerical tests focus on performance of the full *r*BAMG setup process that uses both a least-squares phase to determine interpolation based on q initial random vectors relaxed v times and a subsequent adaptive phase to test and possibly improve the resulting solver. Guided by our convergence estimation process, each iteration of the adaptive phase assesses the current solver's fine-grid convergence factor on random initial guesses for $Ax = 0$ and, when poor convergence is observed, combines the resulting error with the test vectors and errors from earlier adaptive iterations.

In all of our numerical experiments, we used 4 V(1,1) cycles for each adaptive iteration. The values of two threshold parameters used here for convergence estimation are $\rho_{good} = .3$ and $\rho_{bad} = .8$. Note that the adaptive iterations continue until the estimated convergence factor, ρ_{est}, is below ρ_{good} or the estimated total cost for the full solution process is expected to increase. Because of the five-point form of the gauge Laplacian, it is customary for the first coarsening to be red-black, making direct (and 'ideal') operator interpolation possible. We used this coarsening here. Because this produces a 9-point coarse-grid stencil on a rotated uniform coarse grid, we were able to use rotated standard coarsening for the coarser levels. All setup processes and solvers used pointwise lexicographic Gauss-Seidel as the relaxation scheme.

Here we consider a shifted two-dimensional gauge Laplacian (c.f., [16]). The unshifted matrix represents a stencil of the same form as the 5-point Laplacian, but it instead involves unit random complex numbers on the off-diagonal (more precisely, numbers of form $\frac{e^{i\theta}}{h^2}$ on the off-diagonal and $\frac{4}{h^2}$ on the diagonal), and it corresponds to a uniform doubly periodic grid. We, then, shift the matrix so that the smallest eigenvalue is approximately h^2, making the condition number $O(h^{-4})$.

The shifted gauge Laplacian is a very challenging problem on which all classical matrix solvers and conventional multigrid methods are unacceptably slow. Its principle difficulty is that algebraically smooth error tends to have very oscillatory geometric character. This provides a perfect opportunity for adaptive/bootstrap AMG methods to demonstrate their capabilities in the automatic determination of representative smooth error components.

The total costs in terms of work units, the final number of target vectors (test vectors and added adaptive error components), and solver convergence factors are displayed in Tables V-VII for N = 64, 128, and 256, respectively. Each table entry shows the arithmetic average obtained with 10 runs using q different sets of random initial test vectors and ν relaxation sweeps. To compare our *r*BAMG results with an adaptive scheme based on a single test vector, we include a first row in Tables V-VII that shows the results of applying the adaptive AMG scheme (αAMG) developed in [6] to the shifted gauge Laplacian. This adaptive scheme is similar to *r*BAMG with $q = \nu = 1$, except that the number of target vectors is not allowed to increase beyond one in subsequent adaptive iterations. (Any error produced in an adaptive iteration is instead used just to correct the current target vector.) Note that all αAMG convergence factors compare poorly with the optimal results in each table, clearly showing the advantage of *r*BAMG's use of multiple vectors.

Table VIII depicts the best average results observed for each N. Note that the optimal results in terms of work units required definitely show dependence on N. This dependence is to be expected because the weak approximation property becomes more demanding as the problem size grows. That is, the setup phase must produce more accurate test vectors for larger problems because the energy norm of the smoothest vector decreases rapidly relative to the that of the most oscillatory vectors as the problem size grows. It is, however, encouraging that this growth in cost to achieve optimal convergence factors appears to be relatively modest, making this a very viable approach to

DOI: 10.1002/nla

QCD problems that typically involve solving many equations with the same matrix. Reducing or eliminating this dependence on N is the focus of future work.

While the optimal values of q and ν also seem to increase with N, the total cost is fairly insensitive to changes in these parameters. In fact, over the full range of q and ν displayed in these tables, the largest total costs are no more than about twice the optimal cost. Moreover, not unexpectedly, larger q tends to require smaller ν for similar total work, and vice versa. Generally, taking q as the minimum required to ensure well-posedness of the least-squares problems for determining interpolation weights and taking ν = 4 or 5 seems to get us within this optimal regime.

q/ν	1	2	3	4	5	6	7	8	9	10
αAMG	361	293	286	288	305	293	354	299	323	351
1	308 5.00 (.39)	266 4.90 (.31)	286 4.70 (.35)	288 4.10 (.44)	255 4.30 (.33)	256 4.00 (.37)	244 3.90 (.34)	244 3.80 (.35)	244 3.70 (.35)	253 3.80 (.35)
2	358 5.50 (.50)	262 5.40 (.31)	241 4.90 (.32)	244 4.80 (.31)	239 4.50 (.35)	230 4.20 (.35)	220 3.90 (.36)	238 4.20 (.34)	236 4.10 (.34)	219 3.60 (.37)
3	289 6.5 (.35)	236 5.6 (.31)	232 5.4 (.31)	208 4.9 (.30)	222 5.0 (.30)	231 5.0 (.30)	227 4.8 (.32)	222 4.6 (.32)	240 4.8 (.31)	240 4.7 (.32)
4	294 7.3 (.31)	233 6.2 (.33)	214 5.8 (.30)	205 5.5 (.32)	197 5.3 (.30)	192 5.1 (.30)	219 5.4 (.29)	221 5.3 (.30)	230 5.3 (.30)	215 5.0 (.30)
5	331 7.8 (.41)	211 6.6 (.33)	217 6.6 (.31)	227 6.6 (.29)	226 6.5 (.29)	228 6.4 (.28)	215 6.1 (.28)	255 6.5 (.28)	276 6.6 (.29)	253 6.2 (.29)
6	303 8.8 (.31)	242 7.9 (.31)	198 7.1 (.33)	233 7.5 (.28)	221 7.2 (.28)	231 7.2 (.27)	241 7.2 (.27)	261 7.3 (.27)	272 7.3 (.27)	294 7.4 (.28)
7	322 9.4 (.38)	229 8.5 (.30)	215 8.2 (.29)	241 8.4 (.27)	238 8.2 (.28)	266 8.4 (.26)	271 8.3 (.26)	284 8.3 (.26)	277 8.1 (.27)	300 8.2 (.26)
8	327 10.5 (.33)	255 9.6 (.31)	266 9.6 (.29)	242 9.2 (.29)	289 9.5 (.26)	277 9.3 (.25)	273 9.1 (.26)	319 9.4 (.26)	312 9.2 (.26)	318 9.1 (.28)
9	368 11.7 (.31)	250 10.4 (.30)	263 10.4 (.27)	269 10.3 (.26)	283 10.3 (.26)	299 10.3 (.24)	330 10.4 (.25)	320 10.2 (.25)	326 10.1 (.26)	328 10.0 (.25)
10	410 12.8 (.30)	312 11.8 (.30)	294 11.5 (.28)	289 11.3 (.27)	315 11.4 (.24)	310 11.2 (.25)	328 11.2 (.25)	319 11.0 (.24)	393 11.4 (.25)	354 11.0 (.25)

Table V. Total cost (= setup + solver cost), final number of target vectors, and solver convergence factors for *r*BAMG applied to shifted gauge Laplacian and N = 64.

q/ν	1	2	3	4	5	6	7	8	9	10
αAMG	463	519	444	456	366	461	465	379	393	444
1	395 5.40 (.51)	280 5.20 (.30)	278 5.00 (.32)	280 4.90 (.33)	281 4.80 (.33)	290 4.80 (.33)	304 4.00 (.47)	280 4.30 (.39)	277 4.10 (.40)	285 4.10 (.42)
2	399 5.70 (.57)	324 5.40 (.42)	265 5.10 (.36)	255 4.90 (.34)	259 5.00 (.31)	268 5.00 (.33)	274 5.00 (.32)	272 4.80 (.34)	304 4.30 (.44)	291 4.20 (.45)
3	348 6.90 (.38)	264 5.90 (.36)	253 5.60 (.36)	240 5.40 (.32)	249 5.40 (.32)	244 5.20 (.32)	238 5.00 (.33)	247 4.80 (.37)	250 4.80 (.37)	263 5.00 (.33)
4	346 7.2 (.42)	255 6.6 (.31)	244 6.2 (.34)	255 6.3 (.30)	267 6.3 (.32)	244 5.8 (.34)	231 5.5 (.33)	229 5.4 (.31)	236 5.3 (.34)	241 5.3 (.33)
5	301 8.1 (.32)	269 7.5 (.32)	284 7.1 (.40)	240 6.6 (.34)	250 6.7 (.34)	215 6.1 (.34)	233 6.3 (.31)	251 6.4 (.30)	272 6.5 (.33)	262 6.3 (.30)
6	334 8.9 (.39)	290 8.5 (.30)	256 7.8 (.34)	275 7.8 (.34)	252 7.5 (.32)	252 7.4 (.30)	236 7.1 (.29)	265 7.3 (.30)	276 7.3 (.30)	270 7.1 (.30)
7	351 9.9 (.38)	297 9.3 (.30)	272 8.9 (.29)	273 8.7 (.30)	302 8.9 (.28)	252 8.2 (.29)	293 8.5 (.29)	289 8.3 (.29)	291 8.2 (.29)	348 8.6 (.30)
8	375 11.0 (.34)	330 10.4 (.30)	283 9.7 (.33)	321 9.7 (.40)	307 9.6 (.33)	303 9.5 (.28)	275 9.1 (.28)	324 9.4 (.29)	319 9.2 (.30)	308 9.0 (.29)
9	403 12.0 (.33)	376 11.3 (.37)	342 11.0 (.32)	314 1.5 (.36)	321 10.6 (.28)	340 10.6 (.29)	309 10.2 (.28)	363 10.5 (.29)	368 10.4 (.29)	359 10.2 (.29)
10	443 12.7 (.41)	371 12.3 (.29)	366 12.1 (.28)	369 11.9 (.32)	370 11.8 (.28)	379 11.7 (.27)	360 11.4 (.30)	390 11.5 (.28)	356 11.1 (.28)	375 11.1 (.29)

Table VI. Total cost (= setup + solver cost), final number of target vectors, and solver convergence factors for *r*BAMG applied to shifted gauge Laplacian and N = 128.

DOI: 10.1002/nla

q/ν	1	2	3	4	5	6	7	8	9	10
αAMG	443	432	455	440	426	438	417	437	443	408
1	440 6.10 (.50)	365 5.70 (.42)	327 5.20 (.37)	375 4.80 (.47)	338 4.90 (.40)	371 4.40 (.51)	417 3.90 (.60)	367 4.70 (.46)	393 3.90 (.59)	370 4.30 (.49)
2	540 4.80 (.86)	368 6.60 (.37)	365 5.80 (.44)	344 6.00 (.36)	310 5.20 (.38)	316 5.30 (.36)	318 5.00 (.39)	341 4.90 (.41)	318 5.00 (.37)	327 5.00 (.37)
3	445 8.40 (.35)	405 6.70 (.46)	413 6.00 (.56)	363 5.80 (.46)	365 5.70 (.46)	344 5.70 (.42)	310 5.70 (.34)	310 5.50 (.36)	333 5.30 (.39)	327 5.50 (.36)
4	467 8.7 (.43)	370 7.2 (.44)	355 7.1 (.39)	315 6.8 (.34)	315 6.6 (.35)	315 6.0 (.40)	351 6.4 (.39)	370 6.0 (.36)	351 5.9 (.42)	338 5.9 (.39)
5	426 9.3 (.35)	360 7.9 (.40)	347 8.1 (.32)	338 7.4 (.36)	312 7.3 (.33)	331 7.1 (.37)	320 7.1 (.33)	307 6.8 (.33)	327 6.6 (.37)	288 6.3 (.35)
6	468 9.9 (.44)	379 9.2 (.34)	365 9.0 (.30)	371 8.5 (.36)	326 8.2 (.31)	308 7.8 (.33)	310 7.7 (.32)	300 7.4 (.34)	329 7.6 (.34)	348 7.7 (.32)
7	527 11.4 (.39)	393 10.1 (.33)	417 10.2 (.31)	333 9.1 (.33)	335 9.0 (.32)	313 8.6 (.32)	383 8.8 (.37)	342 8.6 (.32)	357 8.6 (.32)	349 8.4 (.33)
8	515 11.7 (.42)	435 11.2 (.32)	427 11.0 (.30)	398 10.5 (.31)	344 9.8 (.32)	346 9.7 (.32)	361 9.7 (.30)	399 9.9 (.30)	348 9.3 (.30)	343 9.1 (.33)
9	602 13.1 (.41)	460 12.1 (.33)	456 11.9 (.32)	392 11.2 (.30)	398 11.1 (.30)	396 10.9 (.31)	387 10.7 (.29)	394 10.6 (.29)	401 10.5 (.31)	377 10.2 (.30)
10	618 13.8 (.45)	530 13.4 (.30)	475 12.8 (.30)	497 12.4 (.38)	402 11.9 (.30)	411 11.8 (.30)	369 11.3 (.32)	411 11.5 (.30)	444 11.6 (.29)	406 11.2 (.29)

Table VII. Total cost (= setup + solver cost), final number of target vectors, and solver convergence factors for *r*BAMG applied to shifted gauge Laplacian and N = 256.

N	64	128	256
Total Cost	192	215	300
(q, ν)	(4,6)	(5,6)	(6,8)
Final number of target vectors	5.1	6.1	7.4
Final convergence factor	.3	.34	.34

Table VIII. Summary of *r*BAMG performance on shifted gauge Laplacian

DOI: 10.1002/nla

6. CONCLUSIONS

We studied the version of *r*BAMG that adds scaled residuals of the target vectors to the least-squares principles for the direct BAMG approach. The general *r*BAMG principle is simply to correct each target vector in the direct least-squares principle by an expression that amounts to applying the relevant relaxation scheme to that vector. Stating the principle in this way shows that *r*BAMG can be applied in cases where the stencil at an F point does not even involve the point itself.

The numerical experiments described in this paper concentrated on the performance of the full setup process that included subsequent adaptive cycles of the current solver used to enhance the initial set of test vectors and thereby improve subsequent least-squares fit of interpolation. An important ingredient of this adaptive approach is the convergence estimation model that we derived. It provided an effective means for judging the convergence quality of the current solver without carrying out the large number of cycles that conventional estimates require. The estimates that this model provided, together with an accurate work estimate we developed, allowed the adaptive scheme to make effective decisions along the way.

DOI: 10.1002/nla

REFERENCES

1. Manteuffel T, McCormick S, Park M, Ruge J. Operator-based interpolation for bootstrap algebraic multigrid. *Numerical Linear Algebra with Applications* 2010; **17(2-3)**:519–537.
2. Brandt A, McCormick S, Ruge J. Algebraic multigrid (AMG) for sparse matrix equations. *In Sparsity and its Applications, D.J, Evans (ed.).*, 1984; 257–284.
3. Ruge J, Stüben K. *Algebraic multigrid (AMG). In Multigrid Methods, vol. 5, McCormick SF (ed.).* SIAM: Philadelphia, PA., 1986.
4. Brandt A. Multiscale scientific computation: Review 2001. *Multiscale and Multiresolution Methods: Theory and Applications (Barth, T.J., Chan, T., and Haimes, R., Eds.)* 2001; :1–96.
5. Brandt A, Brannick J, Kahl K, Livshits I. Bootstrap AMG. *SIAM J. Sci. Comp.* 2011; **33**:612–632.
6. Brezina M, Falgout R, MacLachlan S, Manteuffel T, McCormick S, Ruge J. Adaptive smoothed aggregation (αSA). *SIAM J. Sci. Comp.* 2004; **25(6)**:1896–1920.
7. Brezina M, Falgout R, MacLachlan S, Manteuffel T, McCormick S, Ruge J. Adaptive smoothed aggregation (αSA) multigrid. *SIAM Rev.* 2005; **47(2)**:317–346.
8. Brannick J, Brezina D M Keyes, Livine O, Livshits I, MacLachlan S, Manteuffel T, McCormick S, Ruge J, Zikatanov L. Adaptive smoothed aggregation in lattice QCD. *Lecture Notes in Computational Science and Engineering 55 (2007), O. Wildlund and D. Keyes, eds.* 2007; :505–512.
9. Brezina M, Falgout R, MacLachlan S, Manteuffel T, McCormick S, Ruge J. Adaptive algebraic multigrid. *SIAM J. Sci. Comp.* 2006; **27(4)**:1261–1286.
10. MacLachlan S, Manteuffel T, McCormick S. Adaptive reduction-based AMG. *Num. Lin. Alg. Appl* 2006; **13(8)**:599–620.
11. Brannick J, Frommer A, Kahl K, MacLachlan S, Zikatanov L. Adaptive reduction-based multigrid for nearly singular and highly disordered physical systems. *Electronic Transactions on Numerical Analysis* 2010; **37**:276–295.
12. Brandt A. General highly accurate algebraic coarsening. *Electronic Trans. Numer. Anal.* 2000; **10**:1–20.
13. Brandt A, Brannick J, Kahl K, Livshits I. A least squares based algebraic multigrid solver for Hermitian and positive definite systems. 2009. Unpublished manuscript.
14. Brannick J, Falgout R. Compatible relaxation and coarsening in algebraic multigrid. *SIAM J. Sci. Comp* 2010; **32**:1393–1416.
15. Brandt A. Multi-level adaptive solutions to boundary value problems. *Math. Comp.* 1977; **31**:333–390.
16. MacLachlan S, Oosterlee C. Algebraic multigrid solvers for complex-valued matrices. *SIAM J. Sci. Comput.* 2008; **30(3)**:1548–1571.
17. DeGrand T, DeTar C. *Lattice Methods for Quantum Chromodynamics.* World Sci. Pub. Co., 2006.

DOI: 10.1002/nla