RELAXATION-CORRECTED BOOTSTRAP ALGEBRAIC MULTIGRID (rBAMG)

MINHO PARK*

Abstract. 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 develops another form of BAMG, called rBAMG, that involves modifying the least-squares process by temporarily relaxing on the test vectors at the fine-grid interpolation points.

The basic rBAMG scheme was introduced in an earlier paper [16] 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 important problem in quantum chromodynamics (QCD). While the earlier paper inroduced 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 proves to be much more effective than the conventional approach of just observing successive error reduction factors. Another component of the setup process is the use of the current V-cycle to ensure its effectiveness or, when poor convergence is observed, to expose error components that are not being properly attenuated. 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 developement of an rBAMG algorithm. With this in mind, the results focus on the behavior of rBAMG in terms of the number of initial test vectors used, the number of relaxation sweeps applied to the, and the size of the target matrices.

Key words. iterative methods, multigrid, algebraic multigrid, adaptive multigrid

AMS subject classifications.

1. Introduction.

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

$$Ax = b, \tag{1.1}$$

with only $O(N)$ work, multigrid methods have gained a widespread use for solving the large sparse linear system 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 on a coarser grid, again by relaxation and further elimination on yet coarser grids. Unfortunately, many modern problems involve discontinuous coefficients, complex geometries, and unstructured grids, which inhibit geometric multigrid from being applied.

In contrast to standard multigrid methods, algebraic multigrid (AMG [6, 13, 17]) methods need not take into account any geometric information about the underlying problem. Instead, AMG relies on an algebraic sense of smoothness to construct a hierarchy of matrices and intergrid transfer operators automatically based on the

*Department of Applied Mathematics, Campus Box 526, University of Colorado at Boulder, Boulder, CO 80309-0526. *email:* **parkmh@colorado.edu**

information of the original matrix. 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 [6], the bootstrap AMG approach introduced in [3] and developed further for quantum chromodynamics in [5], an adaptive scheme based on smoothed aggregation (SA) developed in [10] and [11] and developed further for lattice QCD in [7], adaptive AMG schemes developed further in [12], and an adaptive reduction-based AMG algorithm introduced in [14] and [9]. The adaptive approach, as it has usually been implemented, constructs interpolation initially to fit a single test vector, and then tests the resulting 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 constructs interpolation to fit (in a least-squares sense) typically 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 (1.1) 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 is also what is now being studied in the BAMG methods [5]. However, we choose a non-recursive form of the adaptive phase because it facilitates a systematic focus on the other components of the AMG processes.

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. For a compatible relaxation approach to coarse-grid selection, see [2], [4], [5], and[8].

The study of rBAMG 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 rBAMG algorithm studied here is generally fairly effective and robust. Our numerical analysis is thus able to focus on how this scheme behaves numerically in the face of increasing the numbers of test vectors and relaxation sweeps performed on them, as well as the problem sizes.

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 be given by

$$(Pe)_i = \begin{cases} e_i & \text{if } i \in C, \\ \sum\limits_{j\in C_i} w_{ij} e_j & \text{if } i \in F. \end{cases} \tag{2.1}$$

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 remaining grid points 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 $a_{i\ell} \neq 0$). We assume for simplicity in this paper that C_i^c is taken only from the neighborhood of i, so it consists of connected F points and other connected C points that are not in C_i. With this splitting, we obtain

$$a_{ii} e_i \approx -\sum_{j\in C_i} a_{ij} e_j - \sum_{k\in C_i^c} a_{ik} e_k. \tag{2.2}$$

The message here is 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 $\sum_{k \in i} a_{ik} e_k \neq 0$, 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 (2.2) 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

$$e_k \approx \sum_{j \in C_i} \frac{a_{kj}}{\sum\limits_{\ell \in C_i} a_{k\ell}} e_j. \tag{2.3}$$

Substituting this expression into (2.2) and dividing the result by a_{ii} yields interpolation weights given by

$$w_{ij} = \frac{1}{a_{ii}} \left(-a_{ij} - \sum_{k \in C_i^c} \left(a_{ik} \frac{a_{kj}}{\sum\limits_{\ell \in C_i} a_{k\ell}} \right) \right). \tag{2.4}$$

This process of collapsing the unwanted connections in the operator interpolation formula expressed by (2.2) can be viewed as using a crude but properly scaled *truncated* interpolation formula, expressed by (2.3), 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 (2.4).

3. BAMG and *r*BAMG methods. Suppose now that we are given a set of test vectors, $e^{(l)}$, $l = 1, 2, \ldots, q$, 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.) We assume that the coarse-grid interpolatory set, C_i, has already been determined for each F-point i. We also assume 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 [16] for the general case.)

3.1. BAMG. The general form of interpolation operator, P, for AMG is given in (2.1). As described in the previous section, the choice of weights w_{ij} is 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 residuals in a sense that leads to useable 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 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

$$\text{(BAMG)} \qquad \{w_{ij} : j \in C_i\} = \arg\min_{w_{ij}} \sum_{l=1}^{q} (e_i^{(l)} - \sum_{j \in C_i} w_{ij} e_j^{(l)})^2. \tag{3.1}$$

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 rBAMG.

3.2. rBAMG. The least-squares problem for BAMG given by (3.1) can be modified by the addition of the local scaled residual as follows:

$$(r\text{BAMG}) \qquad \{w_{ij} : j \in C_i\} = \arg\min_{w_{ij}} \sum_{l=1}^{q} (e_i^{(l)} - \sum_{j \in C_i} w_{ij} e_j^{(l)} - \frac{r_i^{(l)}}{a_{ii}})^2. \tag{3.2}$$

This change to the fitting process yields a new scheme, which we call rBAMG, that is equivalent to the indirect BAMG (iBAMG) in [16] 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 iBAMG in [16]. 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 temporarily relaxing the equation at point i and then applying the standard BAMG minimization approach. As such, rBAMG is related in spirit to the adaptive relaxation scheme described by Brandt in [1] (and suggested in [2]) that applies relaxation selectively to points exhibiting especially large residuals.

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 these two ingredients. The main idea behind the adaptive process 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, $P^T AP$. On the coarse grid, the pre-images of the fine-grid test vectors under interpolation P (which, for AMG, is 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 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 test can contribute significantly to

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 $\rho_{est} \le \rho_{good}$ 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
```

overall cost, it is best to keep their number as small as possible. To achive this, it is necessary to develop methods that give a estimate of the quality of the current solver that is accurate after only a few applications 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. 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. Using orthogonality of the two components, one can find easily that b_1 is the larger root of the following equation:

$$x^2 - \gamma x + \delta = 0, \tag{4.1}$$

where δ and γ solve

$$\begin{bmatrix} ||e^{(k)}||^2 & -||e^{(k+1)}||^2 \\ ||e^{(k+1)}||^2 & -||e^{(k+2)}||^2 \end{bmatrix} \begin{bmatrix} \delta \\ \gamma \end{bmatrix} = - \begin{bmatrix} ||e^{(k+2)}||^2 \\ ||e^{(k+3)}||^2 \end{bmatrix}.$$

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.

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 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 is needed because fitting interpolation must pay more attention to smooth vectors than it does to oscillatory ones. This is articulated in the so-called weak approximation property that provides the basic principle we use to determine the proper weights used in the least-squares process. See [11] for a discussion of this principle and how it is used in the context of adaptive smoothed aggregation.

5. Numerical experiments.

Our numerical tests focus on performance of the full setup process of rBAMG that uses both a least-squares phase that determines interpolation based on q initial random vectors relaxed v times and a subsequent adaptive phase that aims 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 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.

5.1. Shifted Gauge Laplacian. Here we consider a shifted two-dimensional gauge Laplacian (c.f., [15]). The unshifted matrix represents a stencil of the same form as the 5-point Laplacian, but it instead involves unit random complex numbers in 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 shifted the matrix so that the smallest eigenvalue is h^2, making the condition number $O(h^{-4})$. This is a very challenging problem on which all classical matrix solvers and conventional multigrid methods are unacceptably slow. The principle difficulty with the gauge Laplacian 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 cost in terms of work units, the final number of target vectors (test vectors and added adaptive error components), and solver convergence factor are displayed in Table 1, 2, and 3 for N = 64, 128, and 256, respectively. Each table entry represents an average over 10 runs for a given number of initial test vectors, q, and relaxation sweeps, ν. Table 5.4 depicts the best results observed for each N. Note that the optimal results 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 vectors 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 achive optimal convergence optimal factors apprears to be relatively modest, making this a very viable approach to QCD problem that typically involve solving thousands of equations with the same matirx.

q/ν	1	2	3	4	5	6	7	8	9	10
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 5.1: Total cost (= setup + solver cost), final number of target vectors, and solver convergence factors for rBAMG applied to shifted gauge Laplacian and N = 64.

q/ν	1	2	3	4	5	6	7	8	9	10
4	346	255	244	255	267	244	231	229	236	241
	7.2	6.6	6.2	6.3	6.3	5.8	5.5	5.4	5.3	5.3
	(.42)	(.31)	(.34)	(.30)	(.32)	(.34)	(.33)	(.31)	(.34)	(.33)
5	301	269	284	240	250	215	233	251	272	262
	8.1	7.5	7.1	6.6	6.7	6.1	6.3	6.4	6.5	6.3
	(.32)	(.32)	(.40)	(.34)	(.34)	(.34)	(.31)	(.30)	(.33)	(.30)
6	334	290	256	275	252	252	236	265	276	270
	8.9	8.5	7.8	7.8	7.5	7.4	7.1	7.3	7.3	7.1
	(.39)	(.30)	(.34)	(.34)	(.32)	(.30)	(.29)	(.30)	(.30)	(.30)
7	351	297	272	273	302	252	293	289	291	348
	9.9	9.3	8.9	8.7	8.9	8.2	8.5	8.3	8.2	8.6
	(.38)	(.30)	(.29)	(.30)	(.28)	(.29)	(.29)	(.29)	(.29)	(.30)
8	375	330	283	321	307	303	275	324	319	308
	11.0	10.4	9.7	9.7	9.6	9.5	9.1	9.4	9.2	9.0
	(.34)	(.30)	(.33)	(.40)	(.33)	(.28)	(.28)	(.29)	(.30)	(.29)
9	403	376	342	314	321	340	309	363	368	359
	12.0	11.3	11.0	1.5	10.6	10.6	10.2	10.5	10.4	10.2
	(.33)	(.37)	(.32)	(.36)	(.28)	(.29)	(.28)	(.29)	(.29)	(.29)
10	443	371	366	369	370	379	360	390	356	375
	12.7	12.3	12.1	11.9	11.8	11.7	11.4	11.5	11.1	11.1
	(.41)	(.29)	(.28)	(.32)	(.28)	(.27)	(.30)	(.28)	(.28)	(.29)

Table 5.2: Total cost (= setup + solver cost), final number of target vectors, and solver convergence factors for rBAMG applied to shifted gauge Laplacian and N = 128.

q/ν	1	2	3	4	5	6	7	8	9	10
4	467	370	355	315	315	315	351	370	351	338
	8.7	7.2	7.1	6.8	6.6	6.0	6.4	6.0	5.9	5.9
	(.43)	(.44)	(.39)	(.34)	(.35)	(.40)	(.39)	(.36)	(.42)	(.39)
5	426	360	347	338	312	331	320	307	327	288
	9.3	7.9	8.1	7.4	7.3	7.1	7.1	6.8	6.6	6.3
	(.35)	(.40)	(.32)	(.36)	(.33)	(.37)	(.33)	(.33)	(.37)	(.35)
6	468	379	365	371	326	308	310	300	329	348
	9.9	9.2	9.0	8.5	8.2	7.8	7.7	7.4	7.6	7.7
	(.44)	(.34)	(.30)	(.36)	(.31)	(.33)	(.32)	(.34)	(.34)	(.32)
7	527	393	417	333	335	313	383	342	357	349
	11.4	10.1	10.2	9.1	9.0	8.6	8.8	8.6	8.6	8.4
	(.39)	(.33)	(.31)	(.33)	(.32)	(.32)	(.37)	(.32)	(.32)	(.33)
8	515	435	427	398	344	346	361	399	348	343
	11.7	11.2	11.0	10.5	9.8	9.7	9.7	9.9	9.3	9.1
	(.42)	(.32)	(.30)	(.31)	(.32)	(.32)	(.30)	(.30)	(.30)	(.33)
9	602	460	456	392	398	396	387	394	401	377
	13.1	12.1	11.9	11.2	11.1	10.9	10.7	10.6	10.5	10.2
	(.41)	(.33)	(.32)	(.30)	(.30)	(.31)	(.29)	(.29)	(.31)	(.30)
10	618	530	475	497	402	411	369	411	444	406
	13.8	13.4	12.8	12.4	11.9	11.8	11.3	11.5	11.6	11.2
	(.45)	(.30)	(.30)	(.38)	(.30)	(.30)	(.32)	(.30)	(.29)	(.29)

Table 5.3: Total cost (= setup + solver cost), final number of target vectors, and solver convergence factors for rBAMG applied to shifted gauge Laplacian and N = 256.

6. Conclusions.

We developed rBAMG by simply adding scaled residuals of the target vectors to the least-squares principles for the direct BAMG approach. The general rBAMG 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 rBAMG can be applied in cases where the stencil at an F point does not even involve the point itself.

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 5.4: Summary of rBAMG performance on shifted gauge Laplacian

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 that were 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.

REFERENCES

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

[2] ———, *General highly accurate algebraic coarsening.*, Electronic Transactions on Numerical Analysis, 63 (1992), pp. 521–539.

[3] ———, *Multiscale scientific computation: review 2001.*, In Barth, T.J., Chan, T.F. and Haimes, R. (eds.): Multiscale and Multiresolution Methods: Theory and Application, (2001), pp. 1–96.

[4] A. Brandt, J. Brannick, K. Kahl, and I. Livshits, *A least squares based algebraic multigrid solver for Hermitian and positive definite systems.*, 2009. Unpublished manuscript.

[5] ———, *Bootstrap AMG*, 2010. Unpublished manuscript.

[6] A. Brandt, S. McCormick, and J. Ruge, *Algebraic multigrid (AMG) for sparse matrix equations. in sparsity and its applications, D.J, Evans (ed.).*, (1984).

[7] J. Brannick, D. Brezina, M. Keyes, O. Livine, I. Livshits, S. MacLachlan, T. Manteuffel, S. McCormick, J. Ruge, and L. Zikatanov, *Adaptive smoothed aggregation in lattice QCD*, in Proceedings of DD16, The 16th International Conference on Domain Decomposition Methods, Springer (to appear).

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

[9] J. Brannick, A. Frommer, K. Kahl, S. MacLachlan, and L. Zikatanov, *Adaptive reduction-based multigrid for nearly singular and highly disordered physical systems*, Electronic Transactions on Numerical Analysis, 37 (2010), pp. 276–295.

[10] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation (αSA)*, SIAM J. Sci. Comp., 25(6) (2004), pp. 1896–1920.

[11] ———, *Adaptive smoothed aggregation (αSA) multigrid.*, SIAM Rev., 47(2) (2005), pp. 317–346.

[12] ———, *Adaptive algebraic multigrid.*, SIAM J. Sci. Comp., 27(4) (2006), pp. 1261–1286.

[13] W. Briggs, V. Henson, and S. McCormick, *Multigrid Tutorial*, SIAM, 2000.

[14] S. MacLachlan, T. Manteuffel, and S. McCormick, *Adaptive reduction-based AMG*, Num. Lin. Alg. Appl, 13(8) (2005), pp. 599–620.

[15] S. MacLachlan and C. Oosterlee, *Algebraic multigrid solvers for complex-valued matrices*, SIAM J. Sci. Comput., 30(3) (2008), pp. 1548–1571.

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

[17] J. Ruge and K. Stüben, *Algebraic multigrid (AMG). In Multigrid Methods, vol. 5, McCormick SF (ed.)*, SIAM: Philadelphia, PA., 1986.