AN AGGREGATION MULTILEVEL METHOD USING SMOOTH ERROR VECTORS*

EDMOND CHOW†

Abstract. Many algebraic multilevel methods for solving linear systems assume that the slow-to-converge, or algebraically smooth error is locally constant. This assumption is often not true and can lead to poor performance of the method. Other multilevel methods require a description of the algebraically smooth error via knowledge of the near-nullspace of the operator, but this information may not always be available. This paper presents an aggregation multilevel method for problems where the near-nullspace of the operator is not known. The method uses samples of low-energy error vectors to construct its interpolation operator. The basis vectors for an aggregate are computed via a singular value decomposition of the sample vectors locally over that aggregate. Compared to many other methods that automatically adjust to the near-nullspace, this method does not require that the element stiffness matrices are available.

Key words. preconditioning, multilevel and multigrid methods, smoothed aggregation

AMS subject classifications. 65F10, 65F35, 65F50

1. Introduction.

Algebraic multilevel methods use complementary smoothing and coarse-grid correction processes for solving linear systems from discretized partial differential equations. In these methods, *slow-to-converge error*, or *algebraically smooth error*, is the error that remains after the smoother has been applied and that must be reduced at the next level. The *interpolation* or *prolongation* operator, P, must be able to represent this algebraically smooth error on the coarser level, i.e., P must be constructed such that the algebraically smooth error is in the range of P. To accomplish this, multilevel methods traditionally must make assumptions about the nature of the algebraically smooth error. In standard algebraic multigrid (AMG), algebraically smooth error is assumed to be locally constant or slowly varying along strong couplings [21].

In aggregation multilevel methods for elliptic PDEs, the near-nullspace of the discrete operator is assumed to locally form an approximate basis for the algebraically smooth error. For second- and fourth-order PDEs, these near-nullspace vectors are constants and linear functions [23]; for linear elasticity, these vectors are rigid body modes [9]. If it is known that the solution is physically smooth, then along with the constant vector, the geometric coordinate vectors of the grid points, x, y, and z, and monomial functions of these may locally represent the algebraically smooth error. Similarly, for p-version finite elements, appropriate basis vectors are also known [18].

Algebraic multilevel methods often fail because the above assumptions do not hold. For example, a scaling of the matrix on the right by a non-constant diagonal matrix (different values along the diagonal) will change the near-nullspace. (See [6] for other examples.) In other cases, the near-nullspace and the nature of the algebraically smooth error are simply not known. Further, for aggregation methods, it may be desirable for some problems to use other low-energy vectors in addition to the near-nullspace vectors. A procedure is then needed to compute and incorporate these additional vectors.

*Lawrence Livermore National Laboratory Technical Report UCRL-JRNL-204139. This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

†Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, L-560, Box 808, Livermore, California, 94551 (echow@llnl.gov).

Multilevel methods have been developed that are designed to be more robust and more general by not making assumptions about the algebraically smooth error for a given problem. The earliest methods constructed interpolation to fit global low-energy eigenvectors, which were computed approximately, or approximate solutions to the homogeneous error equation, $Ae = 0$. These methods date back to some of the original AMG papers [20, 21, 19] (see also [6] for additional early references).

Recently developed methods generally fall into two categories: 1) those that attempt to reduce some measure of the interpolation error and usually involve eigenvector calculations for aggregate matrices (also called element agglomerate matrices), and 2) adaptive methods based on approximately solving the homogeneous error equation.

The interpolation operator, P, may be constructed by minimizing a measure of the interpolation error [2, 1, 4],

$$\min_P \left(\max_u \min_v \frac{\|u - Pv\|_2}{\|u\|_A} \right), \quad \forall u \neq 0, \tag{1.1}$$

where it is assumed that A is scaled to have a unit diagonal. A better minimum leads to a better convergence rate for a two-level method. However, to be practical, the minimization (1.1) must be localized over an aggregate of elements or grid points. In an element-based AMG method [4], the interpolation weights are determined essentially by fitting the eigenvectors of the aggregate matrices (which sum to the global matrix, without boundary conditions). For aggregation multilevel methods, a local basis for the algebraically smooth error for an aggregate is formed by the low-energy eigenvectors of the aggregate matrix [13, 7, 4, 10]. Additional basis vectors can be used to improve the convergence rate [7, 10]. The above approaches, however, are designed for finite element discretizations and require access to the element stiffness matrices. (See, however, approaches that avoid the explicit use of element stiffness matrices [14].)

Related to minimizing a measure of the interpolation error is the energy minimization technique [24]. Here, P is constructed by directly optimizing a constant in the subspace correction framework [25], thus improving the convergence rate. The stability and approximation inequalities are satisfied, guaranteeing mesh-independent convergence.

In the category of adaptive methods are techniques that avoid element stiffness matrices and eigenvectors altogether. In these strategies, a few iterations of the "current" method are applied to the homogeneous problem, $Ae = 0$, to reveal the components of the error that are slow to converge. These components can then be used to update the current method. Versions based on both smoothed aggregation [6] and AMG [3, 16, 5] have been developed. In Bootstrap AMG [3], the interpolation weights are constructed via a least-squares fit of a block of algebraically smooth error vectors. The method is currently being extended to high-order PDEs [15]. In αAMG [16, 5], an interpolation formula involving an algebraically smooth error vector is used. Adaptive AMG methods can also be defined when AMG is accelerated by a Krylov subspace method. For example, in a conjugate gradient iteration, the Ritz vectors corresponding to the smallest eigenvalues can be used to update the multigrid method's interpolation operator [17].

In this paper, we present a new method of utilizing samples of algebraically smooth error to build an interpolation operator for the smoothed aggregation method. Like Bootstrap AMG, the interpolation operator is built by fitting a block of algebraically smooth error vectors simultaneously. However, the smoothed aggregation framework which we use requires only one minimization per aggregate, rather than one minimization per fine grid-point. The basis vectors for the aggregates are computed via a truncated singular value decomposition of the sample vectors locally over each aggregate. The method can be made adaptive by approximately solving $Ae = 0$ and updating the set of algebraically smooth error vectors, like the adaptive methods described above.

In Section 2, the new interpolation operator is described. The interpolation is matrix dependent; anisotropies and physical jumps in the PDE problem are reflected in the interpolation operator. The number of sample vectors and the number of basis vectors per aggregate are also discussed. Section 3 presents numerical results on isotropic and anisotropic diffusion problems and a plane strain problem, showing detailed results with different parameters. The section also illustrates the use of a Lanczos process to generate the algebraically smooth sample vectors. Section 4 concludes this paper.

2. Interpolation using algebraically smooth error vectors.

2.1. New interpolation operator. In this section, we describe an interpolation operator based on aggregating variables or grid points that is constructed from samples of low-energy vectors, rather than eigenvectors of aggregate matrices. Here, we assume that disjoint aggregates are given. We also suppose that the coefficient matrix, A, at each grid level has been reordered such that the rows and columns of A for an aggregate are ordered together and consecutively by aggregates. This will simplify the notation in this paper. Finally, we will discuss the two-level case, and defer the extension to multiple levels to Section 2.5.

For aggregate i, let n_i denote the number of variables that are aggregated. Each aggregate will be represented by k_i variables on the coarser level, with $k_i \le n_i$, and each variable corresponding to a basis vector used to interpolate the error over that aggregate.

As in smoothed aggregation [22, 23], we will smooth the tentative interpolation operator in order to reduce the energy of the basis functions. Thus, we begin by seeking a rectangular tentative interpolation operator of the form

$$\tilde{P} = \begin{bmatrix} \tilde{P}_1 & & & & \\ & \ddots & & & \\ & & \tilde{P}_i & & \\ & & & \ddots & \\ & & & & \tilde{P}_J \end{bmatrix}$$

for J aggregates, where $\tilde{P}_i$ is composed of the k_i tentative basis vectors for aggregate i. The matrix $\tilde{P}$ is block diagonal because the grid-point aggregates are disjoint.

Let $S = [s_1, \ldots, s_m]$ be a block of m algebraically smooth error vectors. These vectors may be generated by applying the smoother (to be used in the multigrid solution process) to the homogeneous equations,

$$As_i = 0, \quad i = 1, \ldots, m,$$

(or $AS = 0$ in block form) with a random initial guess for each s_i. The components of the random initial guesses are chosen uniformly from (-1,1). Choosing positive components is very advantageous for some problems, but is less general. (Another method for generating S will be discussed in Section 3.3.) When the near-nullspace of A is known, then S should be composed of these near-nullspace vectors. In this case, method will be the same as the smoothed aggregation method [23] if all k_i equal the number of near-nullspace vectors. The samples, S, may be partitioned as

$$S = \begin{bmatrix} S_1 \\ \vdots \\ S_i \\ \vdots \\ S_J \end{bmatrix}$$

corresponding to the partitioning of $\tilde{P}$.

Given S_i, the portion of the sample vectors corresponding to aggregate i, we seek $\tilde{P}_i$, a rank k_i approximate basis for S_i. Formally, we seek

$$\min_{\tilde{P}_i, W} \|S_i - \tilde{P}_i W\|_2$$

for a given k_i, with $k_i \leq m$. The minimum is achieved when

$$\tilde{P}_i W = U_{k_i} \Sigma_{k_i} V_{k_i}^T,$$

where $U_{k_i} \Sigma_{k_i} V_{k_i}^T$ is the rank k_i truncated singular value decomposition of S_i. By matching variables, $\tilde{P}_i$ is defined to be U_{k_i}, the first k_i left singular vectors. We note that the computations are small, dense SVD computations.

This technique exploits the fact that a local portion (over an aggregate) of an algebraically smooth sample vector may have larger or smaller local energy than the same portion of other algebraically smooth sample vectors. The truncated SVD reduces the effect of the higher-energy portions, while capturing the desirable low-energy components which are more typical. By using more sample vectors than k_i (the minimum required), the constructed bases are generally improved, as will be shown in Section 3.

Recalling (1.1), samples with lower energy must be better approximated by the basis vectors than samples with higher energy, i.e., we wish to have a smaller approximation residual corresponding to samples with smaller energy. This can be accomplished, although imprecisely, by scaling each sample vector, s_j, by $(s_j^T A s_j)^{-1}$ before computing the singular value decomposition. (Empirically this scaling was more effective than $(s_j^T A s_j)^{-1/2}$.) This is particularly important if the sample vectors have very different energy norms. Ideally, we would like to scale the local portion of the sample vectors by their *local* energy norms. However, this would require the construction of aggregate matrices.

Finally, to reduce the energy of the basis functions, $\tilde{P}$ is smoothed one step by a Jacobi smoother to construct the final P. The Jacobi smoother is

$$I - \frac{4}{3\rho} D^{-1} A, \tag{2.1}$$

where D is the diagonal of A, and ρ is the spectral radius of $D^{-1}A$ estimated by a few steps of a Lanczos or Arnoldi method. This smoothing preserves the null vectors of A.

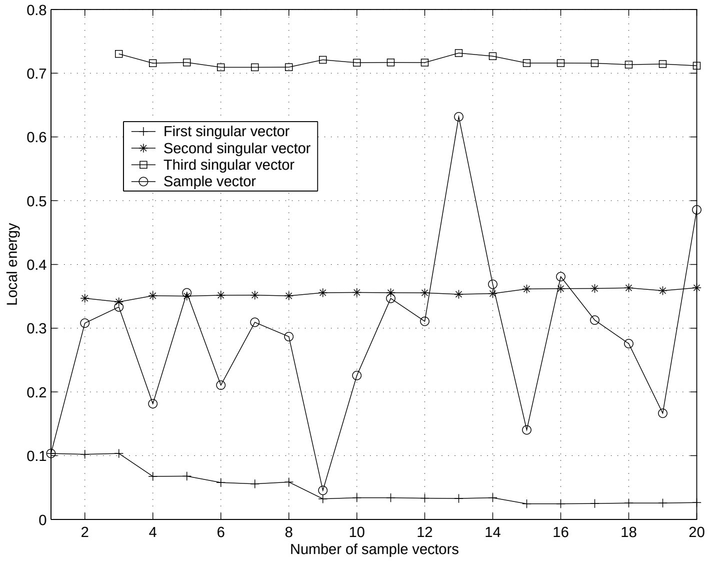


FIG. 2.1. *Local energy of samples and singular vectors vs. number of sample vectors. The energy of the first singular vector generally decreases when more samples are used, but decreases very slowly after a moderate number of samples. Note the decrease in the energy of the first singular vector when low-energy samples are generated at steps 4, 9, and 15.*

2.2. Number of sample vectors. The number of sample vectors is a parameter of this method. A larger number of sample vectors will improve the convergence rate, but this must be balanced with the cost of generating these sample vectors and the cost of larger SVD computations.

Figure 2.1 plots the local energy of each of 20 sample vectors for a 9-node aggregate of a 1-D isotropic diffusion operator. The sample vectors were generated by 3 steps of symmetric Gauss-Seidel (SGS) applied to the homogeneous error equation with random initial guesses. The figure also plots the local energies of the first, second, and third singular vectors as the number of sample vectors increases. It is evident that the first singular vector has the least energy compared to the second and third, and that the energy of the first vector generally decreases when more sample vectors are used. Interestingly, the energy of this singular vector appears to decrease when a new sample vector with very low energy is generated. High-energy sample vectors do not dramatically affect the energy of the first few singular vectors. However, after a moderate number of sample vectors, the energy of the first singular vector then decreases very slowly. Thus, using a very large number of sample vectors is not efficient, and should be balanced with the number of smoothing steps applied to each sample vector. This will also be reflected in the numerical tests of the convergence rate. We note that in these examples, the *local* energy of each sample is plotted. In practice, these local energies cannot be computed if the local aggregate matrices are not known.

2.3. Example. Figure 2.2 shows a sample of algebraically smooth error and three basis vectors produced for an anisotropic diffusion problem with Dirichlet boundary conditions on a 32×32 grid. The direction of anisotropy is 45^o (bottom left to top right). The aggregates, shown by the lighter boundary lines, are 4×4 grid points.

The value at each grid point is indicated by the gray-level of the grid point. For the illustration to be clear, 20 sample vectors were used, with 20 SGS smoothing steps for each vector.

As expected, in Figure 2.2(b), the first basis vector contains nearly locally constant values over each aggregate. Some aggregates have positive values and others negative, which explains the two different gray-levels predominant for this basis vector. It is also noticeable in the figure that at the Dirichlet boundaries there is a decay of the basis vectors toward zero. This problem-dependent behavior is a feature of this method.

Figure 2.2(c) shows that the second basis vector over each aggregate varies slowly in the direction of anisotropy, like the algebraically smooth error, and varies sharply in the cross direction, again like the smooth error. Figure 2.2(d) shows that the third basis vector is oscillatory in the cross direction, which is very helpful in representing the algebraically smooth error in this example. This quadratic-like function would not normally be chosen by other methods; the standard choice of three basis vectors for this example is composed of the constant and the x and y coordinate vectors.

2.4. Number of basis vectors per aggregate. In the smoothed aggregation method [23], the number of basis vectors is chosen beforehand, with each vector being a near-nullspace vector that is known for the problem. In spectral AMGe [10] and the method in [7], the number of basis vectors may be different for each aggregate, and may be chosen such that (1.1) is bounded. In the method presented here, the number of basis vectors may also be different for each aggregate, but no bound on (1.1) is possible.

For aggregate i, the number of basis vectors, k_i, may be chosen based on the singular values from the singular value decomposition of S_i. More basis vectors will improve interpolation for the given aggregate, but increase the cost of the method. One strategy for selecting k_i is to choose the singular vectors that correspond to singular values that are larger than δ times the largest singular value, with $0 \leq \delta < 1$. A sharp decay of the singular values indicates that only a few singular vectors suffice to accurately represent the sample vectors.

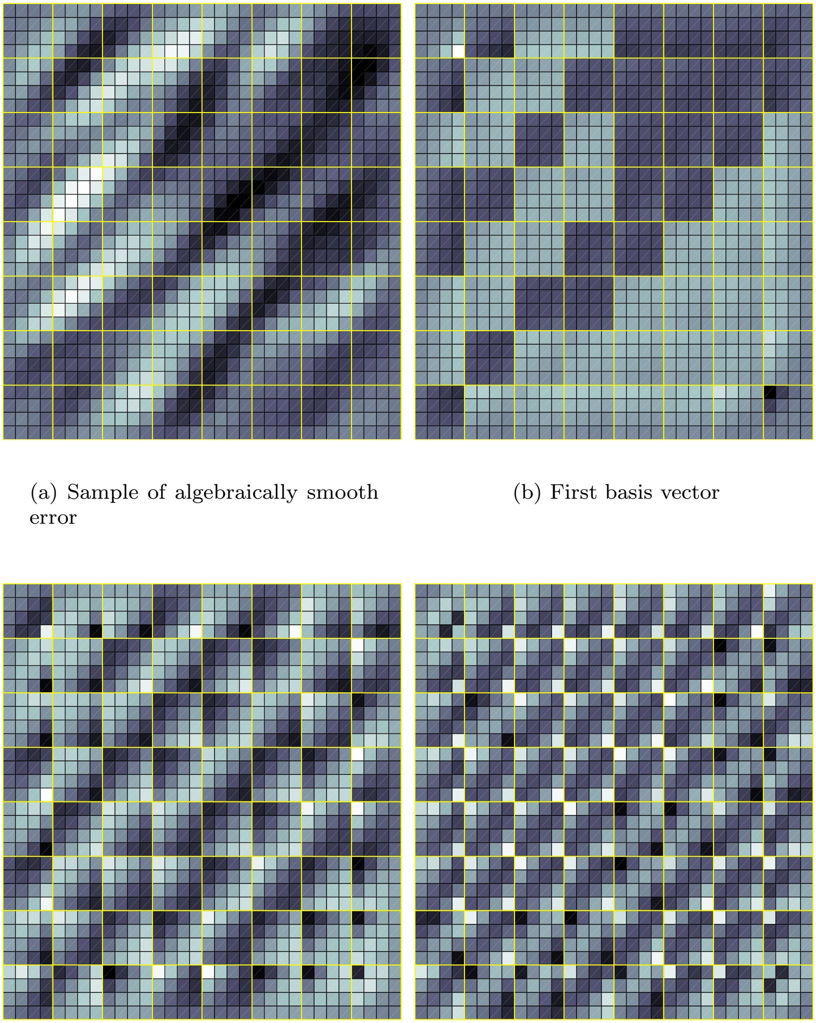
To illustrate this, consider two anisotropic diffusion problems on a 32×32 grid using 4×4 aggregates. The two problems have different angles of anisotropy, 0^o and 45^o. Figures 2.3(a) and 2.3(b) plot the first seven singular values for every aggregate for these two problems. Each curve represents an aggregate, with the singular values scaled such that the largest one for each aggregate is unity. For an angle of anisotropy of 0^o, it is natural to use four basis vectors since there are four lines of grid points in each aggregate, and this is reflected in the singular values. For an angle of anisotropy of 45^o, a smaller number of basis vectors for many aggregates appears adequate.

2.5. Extension to multiple levels. Once P has been defined for an operator A, the operator at the next coarser level may be defined by the Galerkin coarse-grid operator, $A_c = P^TAP$. The method can be extended to multiple levels by applying the method recursively to solve the coarse-grid correction equations involving A_c.

To apply the method to A_c, a set of algebraically smooth error vectors for A_c is needed. These vectors may be generated from scratch, but the following procedure is more effective.

Recall that S denotes m low-energy vectors for A. We seek T, which denotes m low-energy vectors for A_c. If T is constructed such that

$$S = PT,$$



(a) Sample of algebraically smooth error

(b) First basis vector

(c) Second basis vector

(d) Third basis vector

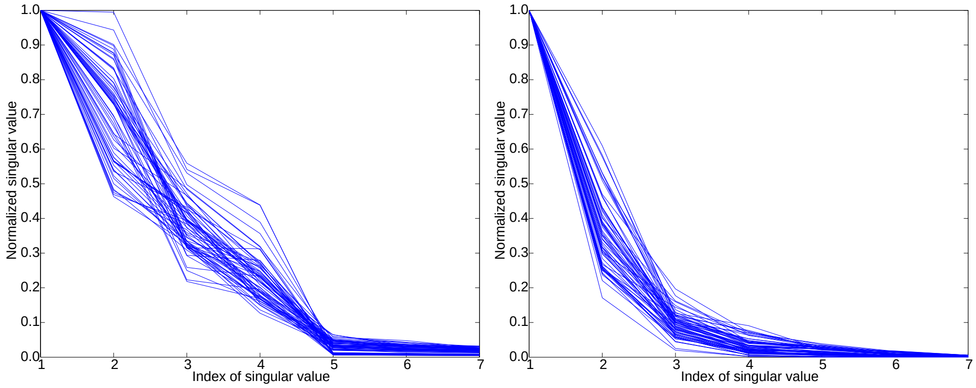
Fig. 2.2. *An anisotropic diffusion problem with Dirichlet boundary conditions. The first basis vectors (b) are nearly constant over each aggregate. The second (c) and third (d) basis vectors vary slowly in the direction of anisotropy, and sharply in the cross direction, like the sample of algebraically smooth error (a).*

then $A_c T = P^T A S \approx 0$ and the vectors in T have small energy. Assuming that $P \approx \tilde{P}$, then aggregate-wise we have

$$S_i \approx \tilde{P}_i T_i, \tag{2.2}$$

where T_i is the portion of T corresponding to aggregate i. From the SVD of S_i already computed,

$$\begin{align} S_i &= U_i \Sigma_i V_i^T \tag{2.3} \\ &\approx \tilde{P}_i \Sigma_{k_i} V_{k_i}^T, \tag{2.4} \end{align}$$



(a) anisotropy angle $= 0^o$

(b) anisotropy angle $= 45^o$

Fig. 2.3. *First seven singular values for each aggregate (each curve represents an aggregate), scaled such that the first singular value equals unity. The plot suggests using four and three basis vectors for (a) and (b), respectively.*

and, by matching variables in (2.2) and (2.4), we define

$$T_i = \Sigma_{k_i} V_{k_i}^T. \tag{2.5}$$

Once T is formed, it generally is not smooth enough due to the approximations made and the fact that this is a representation of a smooth vector on a coarser grid. The energy of the vectors in T can be further reduced by applying a few steps of a relaxation procedure to $A_c T = 0$ using T as an initial guess. We have found that for small problems, only a small number of steps is needed, and additional steps do not significantly improve the convergence rate. For large problems, additional steps can be advantageous. After some experimentation, we used two steps of SGS relaxation, which is a compromise between small and large problems.

For efficiency, it is possible to perform these smoothing steps along with the spectral radius estimation required by the prolongation smoother (2.1) by performing matrix-vector products with a block of vectors simultaneously.

3. Numerical Tests.

In this section, we test smooth-vector interpolation in a multigrid-preconditioned Krylov method. For the multigrid method, restriction is defined to be the transpose of interpolation, the coarse-grid operator is the Galerkin coarse-grid operator, and V(1,1) cycles are used, with symmetric Gauss-Seidel smoothing. Unstructured aggregation of the grid points was performed using the algorithm of the smoothed aggregation method [23], with a strength threshold of 0.08. For coarser levels and non-scalar problems, the degrees of freedom corresponding to a grid point are always aggregated together. We denote this multigrid method by SVMG.

The test problems are discretizations of 2-D unstructured isotropic and anisotropic diffusion equations,

$$\begin{aligned} au_{xx} + bu_{yy} = f \quad &\text{in } \Omega = (0,1)^2 \\ u = 0 \quad &\text{on } \partial\Omega \end{aligned}$$

where $a = b = 1$ for the isotropic problems and $a = 1$ and $b = 1000$ for the anisotropic problems. Random right-hand sides were used. The discretization was linear triangular finite elements. Most of the tests were performed with the largest test problems, UNI7 and ANI7 (isotropic and anisotropic, respectively), which have 205,761 equations and 1,436,481 nonzeros. Smaller test problems were also used. In addition, one test with a plane strain problem was performed and will be described later.

Conjugate gradient acceleration was used with a zero initial guess. The iterations were stopped when the preconditioned residual norm was decreased by 8 orders of magnitude. The experiments were run on a Linux 1.5 GHz Intel Xeon computer with 256 kbytes of cache memory and 512 Mbytes of main memory. Due to the use of randomness in the interpolation, the iteration counts vary by a few steps from test to test.

3.1. Comparison with smoothed aggregation. The smoothed aggregation (SA) method is a powerful method when the near-nullspace of the problem is known. For scalar problems, the practice is to use a single basis vector, the vector of all ones. For UNI7, the PCG-SA method converges in 18 iterations. ANI7 is a much more difficult problem, due to the strong anisotropy on an unstructured grid. Here PCG-SA method converges in 118 iterations.

The new method, SVMG, is designed for the case where the near-nullspace of the operator is not known. When the near-nullspace is known, the smoothed aggregation method using this near-nullspace will perform better. To illustrate this point, we use the UNI7 problem along with a scaled version. Given the original problem matrix, A, the scaled problem is DAD, with $D = \mathrm{diag}(10^{d_1}, 10^{d_2}, \ldots)$ and with the real numbers, d_i, selected randomly from $(0, 6)$. This scaling changes the near-nullspace and is discarded and not known to the SVMG method.

Table 3.1 shows results for UNI7 and its scaled version using SA and SVMG. The table shows PCG iteration counts as well as the time to construct the multigrid preconditioner ("Setup") and the time for the PCG iterations ("Solve"). Five levels were used by the methods. Smoothed aggregation used 3 basis vectors: the constant and the x and y coordinate vectors. SVMG also used 3 basis vectors, constructed from 6 sample vectors, which were themselves generated by 6 SGS relaxation steps. Three basis vectors were used because a single vector might not be adequate when the exact near-nullspace is not known. This, of course, depends on the accuracy of the basis vectors (see also [6], where a single vector is used).

As expected, the results show that for the scaled problem, SVMG performs better than SA. However, for the given parameters, the performance of SVMG on the scaled problem does not match the performance of SVMG or SA on the original, unscaled problem.

The standard basis vectors used in SA for the scaled problem leads to a coarse-grid matrix with sharply varying coefficients. This leads the aggregation procedure to generate many small aggregates at the coarse levels. This in turn leads to very full coarse matrices on subsequent levels and explains the very high setup time for SA on the scaled problem.

3.2. Test results for isotropic and anisotropic problems. Tables 3.2 and 3.3 show SVMG test results with UNI7 and ANI7, respectively, with various numbers of smooth sample vectors and various numbers of SGS smoothing steps per vector. The top portion of each table shows results when a budget of 36 smoothing steps is used to generate the smooth vectors, either using fewer vectors and more smoothing steps per vector, or vice-versa. Three basis vectors were used for each aggregate. Five levels were used in the method.

	UNI7			Scaled UNI7		
	Iterations	Setup	Solve	Iterations	Setup	Solve
SA	12	8.51	3.25	422	117.38	146.28
SVMG	26	11.43	6.76	63	14.90	17.27

Table 3.1

PCG iteration counts and setup and solve timings (in seconds) with SA and with SVMG as the preconditioner for the unscaled and scaled UNI7 problem.

Smooth vectors	Smoothing steps	Iterations	Time (s)		
			Setup	Solve	Total
3	12	41	10.29	10.56	20.85
4	9	31	10.60	7.98	18.58
6	6	26	11.43	6.76	18.19
9	4	28	12.65	7.26	19.91
12	3	31	13.53	8.00	21.53
18	2	39	15.52	10.01	25.53
36	1	59	20.87	15.07	35.94
12	6	23	14.59	6.02	20.61
12	12	16	16.03	4.25	20.28
12	24	14	19.53	3.74	23.27
12	36	12	23.04	3.25	26.29

Table 3.2

PCG-SVMG iteration counts and timings for UNI7 with various numbers of smooth sample vectors and smoothing steps per sample vector.

We express the storage used by the multilevel method in terms of the grid and operator complexities. These terms are common in the AMG literature, e.g., [8]. *Grid complexity* is the total number of degrees of freedom, on all grids, divided by the number of degrees of freedom on the finest grid. *Operator complexity* is the total number of nonzero entries, in all coarse- and fine-grid matrices, divided by the number of nonzero entries in the fine-grid matrix. For the results in Tables 3.2 and 3.3, the grid and operator complexities were approximately 1.4 and 3.6, respectively. For scalar problems, these complexities are considered to be high, since a single basis vector is normally used, rather than three basis vectors as in our case. We use more than one basis vector because multiple basis vectors are readily available (from the SVD computation) and using multiple vectors generally improves the performance of the method when the near-nullspace is only approximated. A study using a smaller number of basis vectors will be given in Table 3.6.

Tables 3.2 and 3.3 show the main point of this paper: given a fixed budget of smoothing steps, it can be worthwhile to use more vectors that are less smooth, than to apply all the smoothing steps to a small number of vectors. When CG acceleration is not used, the effect is much more pronounced. In any code, the improvement depends on the portion of the preconditioner setup cost that is due to generating the smooth vectors. The disadvantage of using too many smooth vectors is the increased cost of the SVD calculations, and this is reflected in the setup timings.

Table 3.4 shows iteration counts for PCG-SVMG, varying the number of smoothing steps applied to T, the block of smooth vectors constructed from (2.5) at each coarse level. For UNI7 and ANI7, the results show no significant improvement if more than 2 smoothing steps are taken. As mentioned, we used 2 steps in for the tests in this paper. For these results, 6 smooth sample vectors and 6 smoothing steps per sample vector were used.

Table 3.5 shows test results for increasing problem sizes, for both the isotropic and

Smooth vectors	Smoothing steps	Iterations	Time (s)		
			Setup	Solve	Total
3	12	221	13.77	58.91	72.68
4	9	187	14.15	49.92	64.07
6	6	164	14.88	43.72	58.60
9	4	158	16.25	42.25	58.50
12	3	156	17.61	41.89	59.50
18	2	165	19.82	44.47	64.29
36	1	198	25.85	53.68	79.53
12	6	126	17.81	33.71	51.52
12	12	109	19.55	29.07	48.62
12	24	97	22.82	25.96	48.78
12	36	86	26.31	23.01	49.32
12	48	80	29.80	21.41	51.21
12	60	76	33.23	20.35	53.58
12	72	68	37.02	18.27	55.29

TABLE 3.3

PCG-SVMG iteration counts and timings for ANI7 with various numbers of smooth sample vectors and smoothing steps per sample vector.

	Number of smoothing steps for T					
	0	1	2	3	4	5
UNI7	51	34	26	24	23	23
ANI7	193	169	164	163	157	153

TABLE 3.4

PCG-SVMG iteration counts for UNI7 and ANI7 when the number of smoothing steps for T at the coarse levels is varied.

anisotropic problems. Again, 3 basis vectors were used for each aggregate, this time constructed from 12 smooth vectors, with each smooth vector generated using 36 SGS relaxation steps. The results show that grid-independent convergence is not present, particularly for the anisotropic problems. It is expected that for larger problems, a larger number of smoothing steps for the sample vectors is required to achieve a convergence rate comparable to that for smaller problems.

Table 3.6 shows test results when a variable number of basis vectors is used for each aggregate. These tests were performed using a MATLAB code which had this functionality, but no timings are available. The test problems were UNI5 and ANI5 (see Table 3.5 for matrix information), and four levels were used in SVMG. We used a block of 12 sample vectors, each constructed from 3 SGS steps. The number of basis vectors was fixed at 1, 2, or 3, or was chosen based on δ (defined in Section 2.4). Table 3.6 shows that a savings in storage can be achieved with a small impact on convergence rate, and in some cases a savings in time may be possible due to a lower operator complexity.

3.3. Using low-energy vectors from the Lanczos method. Instead of using SGS or another smoother to approximately solve the homogeneous error equation, the algebraically smooth error samples can be generated in other ways. For example, starting with a random vector, the Lanczos method can generate approximations to the extremal eigenvalues and eigenvectors of the fine-grid matrix. For SPD problems, it is natural to use the Ritz vectors corresponding to the smallest Ritz values to construct the smooth-vector interpolation operator. If a large number of Lanczos steps are desired, however, the method can be costly both in terms of storage (of the Lanczos vectors) and computation (forming the Ritz vectors).

	Equations	Nonzeros	Levels	Iterations	Time (s)		
					Setup	Solve	Total
UNI4	3321	22761	2	7	0.25	0.03	0.28
UNI5	13041	90321	3	7	1.25	0.11	1.36
UNI6	51681	359841	4	7	5.57	0.49	6.06
UNI7	205761	1436481	5	13	23.12	3.49	26.61
ANI4	3321	22761	2	13	0.24	0.05	0.29
ANI5	13041	90321	3	22	1.30	0.32	1.62
ANI6	51681	359841	4	44	6.24	2.86	9.10
ANI7	205761	1436481	5	89	25.97	23.77	49.74

Table 3.5

Test results for increasing problem sizes. Grid-independent convergence is not present, particularly for the anisotropic problems. It is expected that for larger problems, a larger number of smoothing steps are required to achieve a convergence rate comparable to that for smaller problems.

		Grid complexity	Operator complexity	Iterations
UNI5	$\delta = 0.1$	1.40	3.21	9
	$\delta = 0.2$	1.29	2.17	12
	$\delta = 0.3$	1.20	1.55	16
	$\delta = 0.4$	1.16	1.33	20
	$\delta = 0.5$	1.14	1.25	21
	$k = 3$	1.40	3.03	11
	$k = 2$	1.26	1.90	16
	$k = 1$	1.13	1.22	23
ANI5	$\delta = 0.1$	1.61	6.48	25
	$\delta = 0.2$	1.44	3.86	37
	$\delta = 0.3$	1.37	2.93	47
	$\delta = 0.4$	1.32	2.40	59
	$\delta = 0.5$	1.27	1.97	76
	$k = 3$	1.40	3.03	46
	$k = 2$	1.26	1.90	89
	$k = 1$	1.13	1.22	124

Table 3.6

Test results using a variable number of basis vectors, selected via δ (defined in Section 2.4) or k, the number of basis vectors. As the grid and operator complexities improve by using fewer basis vectors, the iteration counts increase.

We note also that SGS relaxation can be applied to a *block* of vectors, which is computationally very efficient. Block versions of the Lanczos method, on the other hand, generate poorer low-energy Ritz vectors, depending on the block size.

Tables 3.7 and 3.8 show test results using low-energy Ritz vectors for UNI7 and ANI7, respectively. Three basis vectors were used for each aggregate and 5 levels were used in SVMG.

The tables show budgets of either 36 or 72 matrix-vector multiplies. (A budget of 72 corresponds to 36 SGS smoothing steps, but the computational cost is greater.) A different number of low-energy Ritz vectors were constructed. The results show that using more Ritz vectors will improve the convergence rate, and may also improve the total time, depending on the increase in setup cost. In particular, for the ANI7 problem, the best timings are achieved when more than the minimum number of Ritz vectors are constructed.

3.4. Plane strain problem. We briefly consider a problem that strictly requires multiple basis vectors per aggregate, a plane strain problem on a square with homogeneous Dirichlet boundary conditions. The 3 rigid body modes are known

Matvec	Ritz	Iterations	Time (s)		
budget	vectors		Setup	Solve	Total
36	3	50	10.57	12.77	23.34
	4	45	11.24	11.53	22.77
	6	41	12.19	10.50	22.69
	9	36	13.83	9.24	23.07
	12	33	14.73	8.53	23.26
72	3	26	12.11	6.74	18.85
	4	24	12.75	6.24	18.99
	6	21	14.02	5.52	19.54
	9	21	15.63	5.48	21.11
	12	20	17.19	5.26	22.45

TABLE 3.7
PCG-SVMG results for UNI7, using low-energy Ritz vectors.

Matvec	Ritz	Iterations	Time (s)		
budget	vectors		Setup	Solve	Total
36	3	300	15.30	80.69	95.99
	4	289	15.99	77.72	93.71
	6	282	17.30	75.89	93.19
	9	280	18.27	75.72	93.99
	12	281	19.44	75.61	95.05
72	3	204	15.49	54.33	69.82
	4	184	16.51	49.28	65.79
	6	176	17.75	47.12	64.87
	9	173	19.68	46.40	66.08
	12	170	21.22	45.45	66.67

TABLE 3.8
PCG-SVMG results for ANI7, using low-energy Ritz vectors.

from the geometry and grid for this problem and span the near-nullspace of the PDE operator.

The problem was discretized with 217 × 217 linear quadrilateral elements (93,312 equations). The nodes were aggregated regularly, using 3 × 3 node aggregates. Ten smooth vectors were used, with 10 SGS relaxation steps to generate each vector. Three basis vectors were used per aggregate. The multigrid method used 4 levels.

The original problem matrix, A, was generated, as well as a scaled matrix, DAD, with $D = \mathrm{diag}(10^{d_1}, 10^{d_2}, \ldots)$ and with the real numbers, d_i, selected randomly from $(0, 6)$.

Table 3.9 shows the results for the original problem matrix and the scaled matrix, using both the rigid body modes to construct the interpolation operator and the algebraically smooth vectors. Interpolation using the rigid body modes is ideal for the original matrix, but is entirely inappropriate for the scaled matrix. On the other hand, by using the algebraically smooth vectors, both problems can be solved. As expected, however, for the original matrix, the smooth-vector interpolation does not perform as well as interpolating using the rigid body modes.

4. Concluding Remarks.

The smooth-vector interpolation operator presented in this paper is constructed to interpolate samples of algebraically smooth error. The interpolation operator is matrix dependent, but particular entries in the matrix are not needed once the aggregates have been chosen. When the budget for generating the sample vectors is fixed, it can be beneficial to use more samples than the number of basis vectors, even if each sample vector has more energy. This technique is particu-

Poisson	SA		SVMG	
ratio	Original	Scaled	Original	Scaled
.30	13	200+	34	64
.40	18	200+	49	97
.45	24	200+	87	161

Table 3.9

PCG iteration counts for the plane strain problem, using original and scaled matrices. SA or SVMG were compared as the preconditioner. 200+ denotes that no convergence was achieved in 200 iterations.

larly advantageous when many basis vectors are required, since additional vectors are available (from the singular value decomposition) at no additional cost. Compared to other methods such as spectral AMGe [10], this method does not require that the stiffness matrices for each aggregate be available. Generating the sample vectors may also be very efficient, if blocks of vectors are relaxed simultaneously, but this was not tested in this paper.

We note that the performance of the method cannot be better than using the exact near-nullspace vectors if they are known and available. An apparent disadvantage of the method is that for larger problems, smoother sample vectors are required to maintain the same rate of convergence. Thus it may be necessary to use a multigrid method to help generate the sample vectors themselves, as in the adaptive multilevel methods described in the Introduction. SVMG can be made adaptive by approximately solving $Ae = 0$ and updating the set of algebraically smooth error vectors used to construct the interpolation operator. Preliminary results were reported in [11].

We also note that it is also possible to perform the node aggregation using samples of algebraically smooth error, instead of using matrix entries. As in [12], nodes may be aggregated if the samples of algebraically smooth error show a strong coupling between the nodes. Incorporating such a strategy may help develop multigrid methods that are matrix-free.

Acknowledgments. The author wishes to thank Marian Brezina, Robert Falgout, Oren Livne, Scott MacLachlan, Tom Manteuffel, Steve McCormick, Charles Tong, and Panayot Vassilevski for their help in this research and the preparation of this paper. The author is also grateful to two anonymous referees, whose detailed comments added significantly to the presentation and accuracy of this paper.

REFERENCES

[1] J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu, *Convergence estimates for multigrid algorithms without regularity assumptions*, Math. Comp., 57 (1991), pp. 23–45.

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

[3] ———, *Multiscale scientific computation: Review 2001*, in Multiscale and Multiresolution Methods: Theory and Applications, T. J. Barth, T. Chan, and R. Haimes, eds., Springer-Verlag, 2001.

[4] M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid based on element interpolation (AMGe)*, SIAM J. Sci. Comput., 22 (2000), pp. 1570–1592.

[5] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive algebraic multigrid*, SIAM J. Sci. Comput., (submitted).

[6] ———, *Adaptive smoothed aggregation (αSA)*, SIAM J. Sci. Comput., (to appear).

[7] M. Brezina, C. Heberton, J. Mandel, and P. Vanek, *An iterative method with convergence*

rate chosen a priori, Tech. Report UCD/CCM 140, University of Colorado at Denver, April 1999.

[8] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, SIAM Books, Philadelphia, 2000. Second edition.

[9] V. E. Bulgakov, *Iterative aggregation technique for large-scale finite element analysis of mechanical systems*, Computers & Structures, 52 (1994), pp. 829–840.

[10] T. Chartier, R. D. Falgout, V. E. Henson, J. Jones, T. Manteuffel, S. McCormick, J. Ruge, and P. S. Vassilevski, *Spectral AMGe (ρAMGe)*, SIAM J. Sci. Comput., 25 (2003), pp. 1–26.

[11] E. Chow, *An aggregation multilevel method based on smoothed error vectors*, in 2003 International Conference On Preconditioning Techniques For Large Sparse Matrix Problems In Scientific And Industrial Applications, Napa, California, 2003.

[12] E. Chow, *An unstructured multigrid method based on geometric smoothness*, Num. Lin. Alg. Appl., 10 (2003), pp. 401–421.

[13] J. Fish and V. Belsky, *Generalized aggregation multilevel solver*, International Journal for Numerical Methods in Engineering, 40 (1997), pp. 4341–4361.

[14] V. E. Henson and P. S. Vassilevski, *Element-free AMGe: General algorithms for computing interpolation weights in AMG*, SIAM J. Sci. Comput., 23 (2001), pp. 629–650.

[15] O. E. Livne, *Personal communication*, 2004.

[16] S. MacLachlan, *Improving Robustness in Multiscale Methods*, PhD thesis, University of Colorado at Boulder, 2004.

[17] S. MacLachlan, T. Manteuffel, and S. McCormick, *Preconditioning with adaptive AMG*, in 2004 Copper Mountain Conference on Iterative Methods, 2004.

[18] J. Mandel, *Iterative methods for p-version finite elements: Preconditioning thin solids*, J. Comput. Meth. Appl. Mech. Engrg., 133 (1996), pp. 247–257.

[19] S. F. McCormick and J. W. Ruge, *Algebraic multigrid methods applied to problems in computational structural mechanics*, in State-of-the-Art Surveys on Computational Mechanics, A. K. Noor and J. T. Oden, eds., ASME, New York, 1989, pp. 237–270.

[20] J. W. Ruge, *Algebraic multigrid (AMG) for geodetic survey problems*, in Prelim. Proc. Internat. Multigrid Conference, Institute for Computational Studies at Colorado State University, Fort Collins, CO, 1983.

[21] J. W. Ruge and K. Stüben, *Algebraic multigrid (AMG)*, in Multigrid Methods, S. F. McCormick, ed., vol. 3 of Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, 1987, pp. 73–130.

[22] P. Vanek, *Acceleration of convergence of a two-level algorithm by smoothing transfer operators*, Applications of Mathematics, 37 (1992), pp. 265–274.

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

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

[25] J. Xu, *Iterative methods by space decomposition and subspace correction: A unifying approach*, SIAM Review, 34 (1992), pp. 581–613.