CHAPTER 11

Some Remarks on the Hierarchical Basis Multigrid Method

Randolph E. Bank*
Harry Yserentant†

1. Introduction. Originally, multigrid methods were developed for elliptic partial differential equations discretized by difference methods on sequences of uniform and uniformly refined meshes. For most of these problems, multigrid methods are extremely fast solvers for the resulting discrete equations. On the other hand, for many problems of practical interest, uniform meshes are far from optimal. In such cases, finite element methods based on adaptively refined, strongly nonuniform grids are much more appropriate.

It is not obvious how the multigrid method can be effectively applied to problems discretized on nonuniform grids of this type. For many simple decompositions of the final mesh into meshes of different levels, the number of operations per iteration may not be proportional to the number of unknowns in the finest mesh. Indeed, to keep the operation count of optimal order, it is necessary that the number of nodes increase by a constant factor larger than one from one level to the next. As the experience with adaptive local mesh refinement packages like PLTMG [1] [2] shows, this condition is not satisfied for many examples. Often only a few nodes are added at each refinement level. It is therefore necessary to reduce the number of levels artificially in such situations. This can be justified by practical experience [1]. Nevertheless, a complete and sufficiently general theoretical analysis of the convergence behavior in such situations has not yet been given.

The hierarchical basis multigrid method [4] has been developed to overcome these difficulties. In section 2, we formulate the method for general second order boundary value problems in two space dimensions, which are not necessarily self-adjoint and positive definite. We survey some of the properties of the method in sections 3 to 5. In section 6, we present some numerical illustrations using the new version of the package PLTMG [2], which employs the hierarchical basis multigrid method as its linear equation solver.

2. The Hierarchical Basis Multigrid Method. Mathematically, the hierarchical basis multigrid method is a classical block symmetric Gauss-Seidel iteration

* Department of Mathematics, University of California at San Diego, La Jolla, California 92093.
The work of this author was supported by the Office of Naval Research under contract N00014-82K-0197.
† Fachbereich Mathematik der Universität Dortmund, D-4600 Dortmund 50, Federal Republic of Germany.

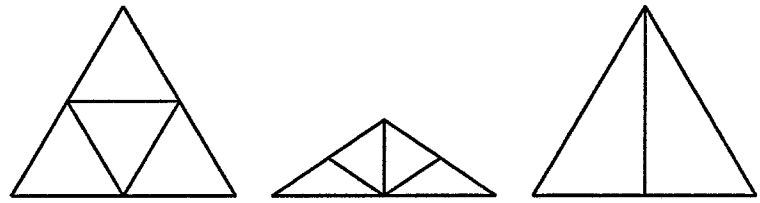

FIG. 1. *Regular refinement (left, center) and irregular refinement(right)*

for solving the discrete boundary value problem, not with respect to the usual nodal basis of the finite element space, but rather with respect to its hierarchical basis. It is closely related to the two-level methods of [3] [7], and the Jacobi-like hierarchical basis method of [9]

In this section we describe the hierarchical basis multigrid method applied to the finite element spaces used in PLTMG [2], but it is straightforward to extend the method to other finite elements, other refinement schemes, and to higher degree polynomial spaces.

Assume that $\mathcal{T}_1$ is an intentionally coarse initial triangulation of a given polygonal domain $\Omega \subset \mathcal{R}^2$. This triangulation is refined several times, yielding a family of nested triangulations $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3 \ldots$. A triangle of $\mathcal{T}_{k+1}$ is either a triangle of $\mathcal{T}_k$ or is generated by dividing a triangle in $\mathcal{T}_k$ into two or four triangles as illustrated in Figure 1.

Refinement into four triangles is called *regular refinement.* If the triangle in question has an obtuse angle, it is refined as in the figure above center; otherwise, it is refined as in the figure above left. Refinement into two triangles is called *irregular refinement.* Irregularly refined elements may not be refined further. This insures that the interior angles of all refined triangles remain bounded away from zero.

The triangles of the initial triangulation $\mathcal{T}_1$ are called level 1 elements and the triangles obtained by the refinement of a level $k-1$ element are called level k elements. In creating the $\mathcal{T}_{k+1}$ from $\mathcal{T}_k$, we allow only the refinement of triangles of level k; this uniquely defines the sequence $\mathcal{T}_k$, $1 \leq k \leq j$. This sequence of meshes does not usually correspond to the actual sequence of meshes generated by an adaptive refinement process like that used in PLTMG, where unrefined triangles of all levels are considered for refinement in every adaptive step. Nonetheless, each fine mesh $\mathcal{T}_j$ generated by such a dynamic process can be uniquely decomposed *a posteriori* into a sequence of triangulations of the type described here. The vertices of the level 1 elements are called level 1 nodes, and the vertices created by the refinement of level $k-1$ elements are called level k nodes. With the definition of $\mathcal{T}_k$ given above, the level k vertices are those generated by the refinement of $\mathcal{T}_{k-1}$.

The triangulations $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \ldots$ correspond to a sequence $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{S}_3 \subseteq \ldots$ of finite element spaces consisting of piecewise linear polynomials. The space $\mathcal{S}_{k+1}$ can be written as the direct sum

$$\mathcal{S}_{k+1} = \mathcal{S}_k \oplus \mathcal{V}_{k+1} \tag{1}$$

where $\mathcal{V}_{k+1}$ is the subspace of $\mathcal{S}_{k+1}$ containing those functions which are zero at all of the nodes of $\mathcal{T}_k$; that is, the vertices of level ℓ for $1 \leq \ell \leq k$. This decomposition leads to the definition of the hierarchical basis for the space $\mathcal{S}_k$. The hierarchical basis for the space $\mathcal{S}_1$ is just the usual nodal basis for $\mathcal{S}_1$. The hierarchical basis for $\mathcal{S}_{k+1}$ consists of the union of the hierarchical basis for $\mathcal{S}_k$ and the nodal basis for $\mathcal{V}_{k+1}$ (the nodal basis functions associated with the level $k+1$ nodes).

For convenience, we fix a finite element space $\mathcal{S} = \mathcal{S}_j$. Formally, the discrete boundary value problem corresponding to $\mathcal{S}$ can be formulated with respect to the hierarchical basis of $\mathcal{S}$. Assume that the hierarchical basis functions are ordered

according to level; that is, corresponding to the decomposition

$$\mathcal{S} = \mathcal{S}_1 \oplus \mathcal{V}_2 \oplus \cdots \oplus \mathcal{V}_j \tag{2}$$

Then the stiffness matrix A corresponding to the hierarchical basis has a block structure given by

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1j} \\ A_{21} & A_{22} & & A_{2j} \\ \vdots & & \ddots & \vdots \\ A_{j1} & A_{j2} & \cdots & A_{jj} \end{bmatrix} \tag{3}$$

where A_{kk} corresponds to the inner products of basis functions in $\mathcal{V}_k$.

We split A into the sum

$$A = L + D + U \tag{4}$$

where L is block lower triangular, D is block diagonal, and U is block upper triangular. Our basic scheme for the solution of the discrete boundary value problem

$$Ax = b \tag{5}$$

is the block symmetric Gauss-Seidel iteration given by

$$x^{(i+1/2)} = x^{(i)} + (D+U)^{-1}(b - Ax^{(i)}) \tag{6}$$

$$x^{(i+1)} = x^{(i+1/2)} + (L+D)^{-1}(b - Ax^{(i+1/2)}) \tag{7}$$

To avoid the expensive direct solution of the $2j$ linear systems involving the diagonal blocks of A, we use inner iterations, and modify (6)-(7) to

$$x^{(i+1/2)} = x^{(i)} + (\tilde{D}+U)^{-1}(b - Ax^{(i)}) \tag{8}$$

$$x^{(i+1)} = x^{(i+1/2)} + (L+\bar{D})^{-1}(b - Ax^{(i+1/2)}) \tag{9}$$

The block diagonal matrices $\bar{D}$ and $\tilde{D}$ correspond to the situation where only the systems involving A_{11} are solved directly, whereas the systems involving the coefficient matrices A_{kk} for $k > 1$ are solved approximately by a fixed number of inner iteration steps. The theory developed in [4] allows:

- a single backward Gauss-Seidel step for each diagonal system in (8) and a single forward Gauss-Seidel step for each system in (9) (except those involving A_{11}).
- an arbitrary fixed number of symmetric Gauss-Seidel steps for all systems (except those involving A_{11}) arising in both half steps (8)-(9).

In PLTMG [2], one symmetric Gauss-Seidel is used for the inner iteration for self-adjoint problems. For strongly nonsymmetric problems, it is desirable to damp the inner symmetric Gauss-Seidel iteration. The resulting iteration is like SSOR except that the relaxation parameter ω is less than one.

3. Convergence Results. A complete convergence theory for this process in the case of self-adjoint, positive definite, second order elliptic boundary value problems in two space dimensions has been given in [4]. In this case, $U = L^t$, $D = D^t$, $\bar{D} = \tilde{D}^t$, and the iteration (8)-(9) can be written as

$$x^{(i+1)} = x^{(i)} + B^{-1}(b - Ax^{(i)}) \tag{10}$$

with a symmetric and positive definite matrix B given by

$$B = (L + \tilde{D})^t(\tilde{D} + \tilde{D}^t - D)^{-1}(L + \tilde{D}) \tag{11}$$

The optimal estimate for the speed of convergence of this iteration with respect to the energy norm induced by the matrix A is

$$\|x^{(i+1)} - x\| \leq (1 - \kappa^{-1})\|x^{(i)} - x\| \tag{12}$$

In (12), x denotes the exact solution of the linear system (5), $\|x\|^2 = x^t A x$ is the energy norm, and κ is the spectral condition number of the preconditioned matrix

$$B^{-1/2} A B^{-1/2} \tag{13}$$

Using only very weak assumptions, which are almost always satisfied in practice, we were able to show that

$$\kappa = O(j^2) \tag{14}$$

where j is the number of refinement levels. In particular, we assume shape regularity of the finite elements, but do not require that the global mesh be quasiuniform. No global regularity of the problem is used (beyond the $\mathcal{H}^1$ regularity necessary to define the weak form), and only local ellipticity constants enter into the proofs.

This result means that only $O(j^2|\log\epsilon|)$ iterations are required to reduce the energy norm of the error by a factor of ϵ. This can be improved to

$$O(j\log\epsilon) \tag{15}$$

iteration steps by using a conjugate gradient or minimum residual acceleration scheme in conjunction with the basic iteration.

4. Implementation. Algorithmically, the hierarchical basis stiffness matrix (3) should not be assembled and stored explicitly. As we have shown in [4], and have implemented in PLTMG [2], the iteration (8)-(9) can be realized in a fashion similar to a standard multigrid V-cycle [5] using a point symmetric Gauss-Seidel (or SSOR) smoother. The difference is that, on a given level k, for the hierarchical basis method only those unknowns corrresponding to the space $\mathcal{V}_k$ are smoothed, as opposed to the unknowns associated with all the vertices in $\mathcal{T}_k$, as in the standard multigrid method.

Therefore only the (sparse) diagonal blocks A_{kk} for $1 \leq k \leq j$ need to be stored, along with certain entries allowing one to generate the product $A_{ik}y$ for $i \neq k$. The details of this scheme are described in [4]. In any event, the total number of floating point numbers required for this implicit representation of the stiffness matrix is $9N + O(1)$ for the nonsymmetric case, and $5N + O(1)$ for the symmetric case, where N is the dimension of $\mathcal{S}$. This should be compared with $7N + O(1)$ and $4N + O(1)$, respectively, for the standard nodal stiffness matrix.

In the same fashion, one can show that the number of operations necessary to perform one iteration is $O(N)$, regardless of the distribution of the nodes among the levels. In view of the results cited in section 3, at least for symmetric, positive definite problems, a total of

$$O(jN\log\epsilon) \tag{16}$$

operations are required to reduce the energy norm of the error by a factor of ϵ. Practically, this represents a logarithmic-like growth in the number of operations as a function of the number of unknowns. This estimate is slightly suboptimal when compared to standard multigrid methods applied to sufficiently regular problems with geometrically increasing subspace dimensions, where the corresponding work estimate would be

$$O(N\log\epsilon) \tag{17}$$

Unlike the usual multigrid methods, however, (16) requires only the weak assumptions of sections 2 and 3 for the continuous problem and for the sequence of meshes.

5. A Counterexample. Both a standard multigrid V-cycle, using a point Gauss-Seidel smoother, and a cycle of the hierarchical basis multigrid method can be interpreted as a sequence of one-dimensional corrections. Compared with the standard multigrid method, in the hierarchical basis multigrid method certain directions are skipped. As a curiousity, we present a simple example that this does not necessarily mean slower convergence.

Consider the one-dimensional boundary value problem

$$-u'' = f \qquad 0 < x < 1 \tag{18}$$

$$u(0) = u(1) = 0 \tag{19}$$

discretized by piecewise linear finite elements on the grids

$$x_i = i2^{-k} \qquad i = 0, \ldots, 2^k \tag{20}$$

for $k = 1, 2, \ldots, j$. If we first smooth the unknowns associated with odd indices, and then those with even indices, the standard V-cycle is not an exact solver. On the other hand, the discretization matrix for this boundary value problem with respect to the hierarchical basis is a diagonal matrix. Therefore, the hierarchical basis multigrid method, smoothing only the unknowns associated with the odd indices on every level, is an exact solver.

6. Numerical Results. As mentioned above, the hierarchical basis multigrid method is the linear solver used in the new version of PLTMG [2]. This package was used to solve the problems presented in this section.

The first example is the Helmholtz equation

$$-\Delta u - 100u = 1 \tag{21}$$

on the unit square $\Omega = [0,1]^2$. The boundary conditions

$$u = 0 \tag{22}$$

are imposed on $\partial\Omega$. The eigenvalues of the differential operator associated with this boundary value problem are

$$\lambda_{k\ell} = (k^2 + \ell^2)\pi^2 - 100 \tag{23}$$

for $k, \ell = 1, 2, \ldots$. The corresponding eigenfunctions are

$$u_{k\ell}(x, y) = \sin(k\pi x)\sin(\ell\pi y) \tag{24}$$

This means that four eigenvalues are negative and the corresponding eigenfunctions form a six-dimensional space. To obtain meaningful results, the triangulation associated with the first level in the hierarchical basis multigrid method must be fine enough to represent these eigenfunctions. We began with a uniform 9×9 grid. PLTMG was used to adaptively refine this mesh, obtaining a fine mesh with 517 points distributed among three levels. Both grids, together with the final solution, are shown in Figure 2.

In the table below we summarize the numerical results obtained using the hierarchical basis multigrid method in conjunction with conjugate gradient acceleration.

iteration	1	2	3	4	5	6	7	8	9
digits	-0.013	0.91	2.20	2.85	3.33	3.44	3.50	3.60	4.74

The number of correct digits is defined by

$$digits = -\log_{10}(\|x^{(i)} - x\|/\|x\|) \tag{25}$$

where the norm is the continuous $\mathcal{H}^1$ norm of the functions represented by the given coefficient vectors, and where x is the exact solution of the discrete equation. The initial guess $x^{(0)}$ was zero.

The average rate of convergence is approximately 0.3, which is not very different from the rate of convergence observed in [4] for a boundary value problem involving the

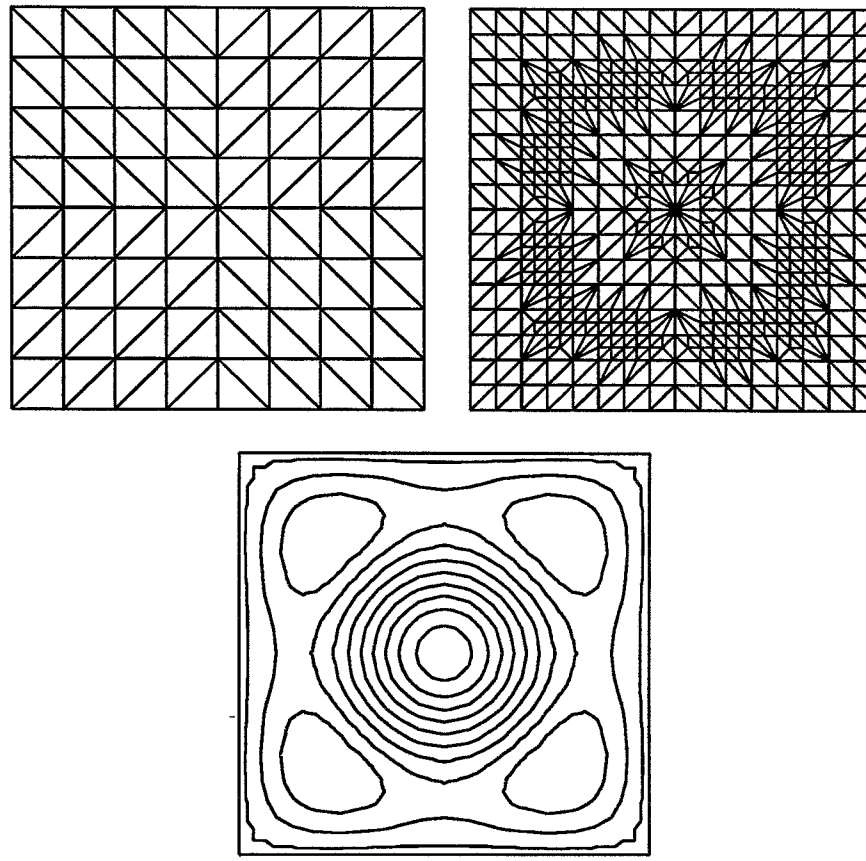

FIG. 2. *The coarse mesh (top left), refined mesh (top right), and the solution (bottom) for (21)*

Laplace operator. With coarser initial triangulations, the convergence rate decreases, at least in the initial steps. For example, using the same fine grid, but using four levels and a coarse 5×5 grid, the convergence rate is 0.5. For five levels, using a 3×3 coarse mesh, the convergence rate is 0.84. This behavior is typical of many multigrid procedures, and is explained by poor approximation properties of the coarsest mesh. The 3×3 mesh, for example, had only one interior point (one unknown), and therefore could not hope to give a satisfactory approximation of the six-dimensional space corresponding to the negative eigenvalues.

Our second example is the convection-diffusion problem

$$-\Delta u + \beta \cdot \nabla u = 1 \tag{26}$$

with $\beta^t = (100, 100)$. The region Ω is the unit square, and the boundary conditions are the same as in the first example. This is a moderately difficult convection dominated problem with sharp boundary layers. For problems like this, PLTMG uses a stabilized version of the finite element method, the streamline diffusion method [6]. We started with the same coarse grid as in the first example, and ended with the highly nonuniform final grid shown in Figure 3.

This grid has 502 vertices distributed among five levels. The discrete equations have been solved using a generalized minimum residual acceleration procedure similar

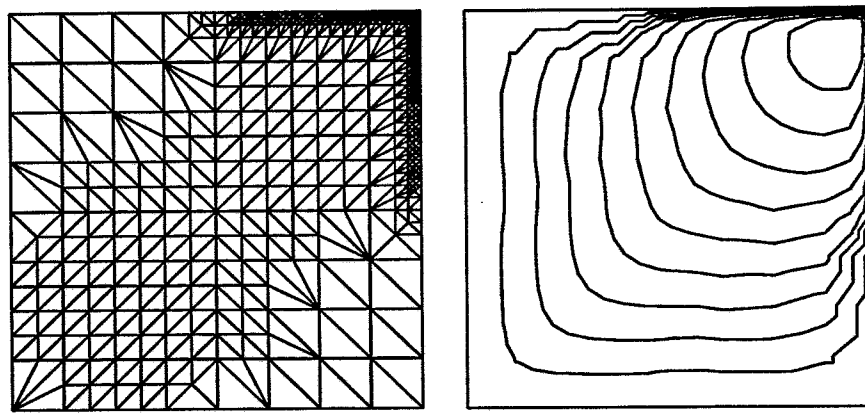

FIG. 3. *The refined mesh (left) and the solution (right) for (26)*

to orthomin [8], starting from an initial guess of zero. The results are summarized below:

iteration	1	2	3	4	5	6	7	8	9	10
digits	0.43	0.79	0.97	1.62	2.00	2.49	2.78	3.27	3.73	4.17

The average rate of convergence was approximately 0.38. These results demonstrate that the hierarchical basis multigrid method is a reasonable solver for this class of problems too. As in the first example, there is sensitivity to the level of approximation of the coarsest mesh. Solving the same problem using a 5×5 coarse mesh and six levels led to a convergence rate of 0.76; the same rate was observed using a 3×3 coarse grid with seven levels.

The solver in PLTMG adaptively determines how many previous directions should be used for orthogonalization in the acceleration procedure. The criterion is based on the observed rate of reduction in the ℓ^2 norm of the residual. In solving the problem using the 9×9 coarse grid, the minimum residual procedure orthogonalized the current direction with respect to only one previous direction. The same was true for the 5×5 case, except for one iteration where two previous directions were used. For the 3×3 case, up to seven previous directions were used; four to six directions were typical. Thus, although the convergence rate was nearly the same for the 5×5 and 3×3 cases, the acceleration procedure worked much harder to achieve that rate in the latter case.

REFERENCES

[1] R. E. BANK, *PLTMG Users' Guide, Edition 4.0*, Tech. Rep., Department of Mathematics, University of California, San Diego, 1985.

[2] R. E. BANK, *PLTMG Users' Guide, Edition 5.0*, Tech. Rep., Department of Mathematics, University of California, San Diego, 1988.

[3] R. E. BANK AND T. F. DUPONT, *Analysis of a Two Level Scheme for Solving Finite Element Equations*, Tech. Rep. CNA-159, Center for Numerical Analysis, University of Texas at Austin, 1980.

[4] R. E. BANK, T. F. DUPONT, AND H. YSERENTANT, *The hierarchical basis multigrid method*, Numer. Math., (1988).

[5] W. HACKBUSCH, *Multigrid Methods and Applications*, Springer-Verlag, Berlin, 1985.

[6] C. JOHNSON, *Finite element methods for convection-diffusion problems*, in Computing Methods in Applied Sciences and Engineering, North Holland, Amsterdam, 1982, pp. 311–324.

[7] J. F. MAITRE AND F. MUSY, *The contraction number of a class of two-level methods; an exact evaluation for some finite element spaces and model problems*, in Multigrid Methods (Lecture Notes in Mathematics 960), Springer-Verlag, Berlin, 1982, pp. 535–544.

[8] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimun residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[9] H. YSERENTANT, *On the multi-level splitting of finite element spaces for indefinite elliptic boundary value problems*, SIAM J. Numer. Anal., 23 (1986), pp. 581–595.