SIAM J. SCI. STAT. COMPUT.
Vol. 4, No. 2, June 1983

0196-5204/83/0402-0009 $01.25/0

MULTIGRID METHODS FOR DIFFERENTIAL EIGENPROBLEMS*

A. BRANDT†, S. McCORMICK‡ AND J. RUGE‡

Abstract. This paper develops an efficient multigrid algorithm for solving the eigenvalue problem associated with a linear differential operator. The algorithm is based on the full approximation scheme (FAS) and incorporates a Ritz projection process for simultaneous computation of several eigenvalues and their eigenvectors. Included are the results of some numerical experiments that illustrate its performance in various contexts.

Key words. multigrid, eigenvalue, nonlinear, partial differential equations, eigenvectors, eigenfunction

1. Introduction. The usual method for finding eigenvalues of a differential operator is to discretize the problem and solve the resulting matrix eigenvalue problem by some algebraic technique. If, as we assume here, the first several eigenvalues and corresponding eigenvectors are desired, then one may use a simultaneous or block version of such methods as inverse iteration, Rayleigh quotient iteration or Lanczos.

This approach of treating the discrete problem as a purely algebraic one can result in a loss of valuable information, especially concerning the smoothness of the eigenvectors. In general, the operator's eigenvectors corresponding to the desired smaller eigenvalues are very smooth, so that they are fairly well approximated on coarser grids. Certain multigrid processes (e.g., FMG described in § 3.2) take full advantage of this smoothness and are therefore very effective for solving such problems.

The experiments we have performed indicate that the first eigenvalue of a differential operator can be approximated to within truncation error with a little more work than is needed for solving the related boundary value problem by multigrid. When more than one eigenvalue is desired, the work needed per eigenvalue increases somewhat due primarily to the orthogonalization and Ritz steps used to prevent all of the emerging eigenvalue approximations from converging to the first eigenvalue. Nevertheless, as we note in § 7, the total work is $\frac{20}{3}q^2n + O(q^2n + q^3 \log n)$, where n is the number of fine grid points and q is the number of desired eigenvalues.

After introducing the notation and some basic multigrid ideas, the method for finding an approximation to the first eigenvalue of the operator is detailed and discussed. This method is then extended in § 6 to the computation of several eigenvalues.

Basically, the algorithm proposed in this paper uses the version of multigrid that treats the eigenvalue problem as a nonlinear problem on all grids. The problem is solved on successively finer grids, using the solution at each level as the initial guess for the next. To improve this initial guess, a multigrid cycle is then performed for each eigenvector, retaining nonlinearity on coarser grids and maintaining separation of the vectors by coarse-grid orthogonalization with respect to previous eigenvectors.

The Ritz projection is used to maintain a stable basis for the emerging invariant subspace approximation and results in accelerating the speed of convergence of the multigrid iteration to the true eigenvectors. It raises several questions in algorithm design, as we shall see.

---

* Received by the editors October 6, 1981. This work was supported by the National Science Foundation under grant MCS80-17056 and the Air Force Office of Scientific Research under grant AFOSR-F-33615-79-C-3223.

† Applied Mathematics Department, Weizmann Institute of Science, Rehovot, Israel.

‡ Department of Mathematics, Colorado State University, Fort Collins, Colorado 80523.

Several other algorithms incorporating multigrid methods have been proposed for the solution of differential eigenvalue problems. Hackbusch [5] developed a method for approximating the eigenvalues of an elliptic differential operator. The basic algorithm is similar to ours although the emphasis of his work is mainly theoretical. Strakhovskaya [9] proposes another method for approximating the first eigenvalue that is similar to Hackbusch's but that uses coarser grids to solve the residual equation obtained on the finest. Clemm and Guderley [4] use the theory of a two-level method for linear problems to develop a method for finding several eigenvalues. It is much different from the modern recursive multigrid approach taken here. McCormick [7] uses Newton's method (that is, Rayleigh quotient iteration) with multigrid as the inner-loop equation solver, together with Ritz projections to compute several eigenvalues and eigenvectors.

The method developed by Alcouffe, et al. [1] for solving linear problems has been extended to apply to eigenvalue problems. It is similar to our approach for computing the first eigenvalue, although their emphasis is in applying the method to their problem and ours is in a full development of the algorithm.

The essential features of our algorithm were developed by the authors in 1979. In fact, the present algorithm is an improvement of the one coded and distributed on magnetic tape at the 1979 multigrid workshop at Yale University.

2. Notation. Let L be a differential operator on a set of functions defined on a domain Ω in $\mathbb{R}^d$. Let $G^1, G^2, \cdots, G^m$ be a sequence of increasingly finer grids that extend over Ω. Assume that all grids are uniform and that each, except G^1, is a refinement of the previous grid made by halving the mesh size. Let L^k denote the finite difference approximation to L on grid G^k.

Since interaction between grids is necessary, we need a procedure for transferring functions from one grid to another. Let G^k and G^l be two different grids and let u^k be a function defined on G^k. Let I_k^l be a mapping from the functions on G^k to the functions on G^l such that:

a) if G^k is coarser than G^l, $I_k^l u^k$ is the function obtained on G^l by linear interpolation of u^k to G^l; and

b) if G^k is finer than G^l, the value of $I_k^l u^k$ at a given point of G^l is the weighted average of a small number of points neighboring the corresponding G^k point. (This includes the frequently used definition that the value of $I_k^l u^k$ at a given point of G^l is just the value of u^k at the corresponding point of G^k.)

For $k < l-1$ we assume that $I_l^k = I_{l-1}^k I_l^{l-1}$.

In the following, the inner product denoted by $\langle \cdot, \cdot \rangle$ is a discrete approximation to the continuous $L_2(\Omega)$ inner product which, in d dimensions, is given by

$$\langle u^k, v^k \rangle = \sum_{x \in G^k} h_k^d u^k(x) v^k(x).$$

Here, h_k is the mesh size of G^k and $u^k(x)$ and $v^k(x)$ represent the values of u^k and v^k at the grid point x.

3. Basic multigrid processes.

3.1. Multigrid cycle. The efficiency of multigrid methods results from the fact that, although relaxation is usually slow to converge, it is quick to reduce high-frequency error components. This allows the problem to be transferred to a coarser grid where the error can be resolved with much less work. (Not only is relaxation cheaper per sweep on coarser grids, but the solution process is also much more

effective.) The coarse grid equation can be solved by relaxation and appeal to still coarser grids. The coarsest grid used is chosen so that solution of the problem there is inexpensive compared to the work performed on the fine grid. The number of relaxation sweeps needed to smooth the error on each grid is generally small and can be predetermined by the usual mode analysis, for example. The process of using a predetermined number of sweeps per grid is called fixed cycle multigrid. Such a cycle is defined by the following steps (where G^l is the finest grid).

1. Set $k \leftarrow l$.
2. Relax ν_1 times on the G^k problem.
3. If $k = 1$, go to 4.
   Otherwise, set $k \leftarrow k-1$, transfer the problem to a coarser grid and go to 2.
4. Relax ν_2 times on the G^k problem.
5. If $k = l$, stop.
   Otherwise, correct the G^{k+1} solution approximation using the solution of the G^k problem, set $k \leftarrow k+1$ and go to 4.

(This actually illustrates the so-called V-cycle, where one multigrid cycle on the G^k problem involves exactly one multigrid cycle on G^{k-1}. The so-called W-cycle uses two G^{k-1} cycles before correcting the G^k solution.) Note that the problem on G^1, the coarsest grid, is solved by $\nu_1+\nu_2$ relaxation sweeps.

3.2. Full multigrid (FMG) algorithm. The cycling scheme described above does not provide a procedure for determining a good initial guess for the problem on G^l. However, since the G^l problem is approximated by one on G^{l-1}, it is natural to determine a good starting vector by solving the G^{l-1} problem and interpolating the result to G^l. This suggests solving the problem first on G^1 (by some convenient method) and then solving the grid G^l problem by multigrid cycling, using the interpolated solution from level $l-1$ as the initial approximation ($l = 2, 3, \cdots, m$). Cubic interpolation may be used here for the initial approximation to the G^l cycle (cf. [3]). Algorithms utilizing this procedure are referred to by the term full multigrid (FMG).

Throughout this paper, l denotes the currently finest level, that is, the finest level reached thus far in the FMG procedure, and level k is the current level in the cycling process.

3.3. Full approximation scheme (FAS). For linear problems, the residual equation for the G^k problem may be transferred to G^{k-1}. The solution of the resulting coarse grid problem then yields an approximation to the error of the G^k solution approximation, so the G^{k-1} solution is interpolated to G^k and added to the solution there. However, for nonlinear problems, such as eigenproblems, instead of the actual error, it is more convenient to approximate the fine grid solution itself on coarse grids. This leads to the very powerful but subtle so-called full approximation scheme (FAS), the details of which can be found in [2]. The essence of FAS is the construction of the coarse grid problem so that its solution is a good approximation to the fine grid solution transferred to the coarse grid. This ensures that the difference between the fine grid approximation and the coarse grid solution is an approximation to the smooth components of the fine grid error so that it can be used as a correction to reduce these components. Note, at convergence, that the coarse grid solution must provide a zero correction; that is, the coarse grid solution is just the fine grid solution transferred to the coarse grid.

To be more specific, assume that FMG is applied to $LU = f$ and level l is the currently finest grid. The G^l version of the problem is written as $L^lU^l = f^l$. Assuming

we have an approximation, u^l, to the exact solution, U^l, and the error $U^l - u^l$ is smooth, then the problem is transferred to G^{l-1} as

$$L^{l-1}U^{l-1} = I_l^{l-1} f^l + \tau^{l-1},$$

where $\tau^{l-1} = L^{l-1} I_l^{l-1} u^l - I_l^{l-1} L^l u^l$. (Note that τ^{l-1} is approximately what is needed to obtain $I_l^{l-1} U^l$ as the coarse grid solution; that is, if $u^l = U^l$, then the level $l-1$ solution is exactly $I_l^{l-1} U^l$.) Transfer of the problem to grid G^k, $k \leqq l-1$, using the same idea yields the problem

$$L^k U^k = I_l^k f^l + \tau^k, \tag{3.1}$$

where

$$\tau^k = L^k I_{k+1}^k u^{k+1} + I_{k+1}^k (\tau^{k+1} - L^{k+1} u^{k+1}). \tag{3.2}$$

The notation we adopt here uses the usual somewhat simplified multigrid conventions. However, this simplicity introduces unfortunate ambiguities. Specifically, note that capital letters are used to denote exact solutions and small letters to denote current approximations to these solutions. Thus, U^k is the exact solution to the G^k problem, so it depends on τ^k, which in turn depends on $u^{k+1}, \cdots, u^l$, and l itself. Thus, τ^k and hence U^k do not represent the same functions throughout the solution process. This is an ambiguity that is necessary to avoid substantially more complex notation, but should cause no difficulty to the reader if this dependence of the coarse grid solution on the emerging fine grid solution is kept in mind.

Once a suitable approximation, u^k, to U^k is found, the approximation u^{k+1} can be corrected according to

$$u^{k+1} \leftarrow u^{k+1} + I_k^{k+1} (u^k - I_{k+1}^k u^{k+1}). \tag{3.3}$$

This is the so-called FAS interpolation step. (Note that I_{k+1}^k is used in two different terms in (3.2). In some instances, it is advisable to use different interpolation schemes for each of these terms, but it is essential that the interpolation used for u^{k+1} in (3.2) be identical to the interpolation used in (3.3).)

The processes mentioned above form the basis for the algorithms used in this paper.

4. The problem. The problem treated in this paper is to find approximations to the first few eigenvalues and associated eigenvectors of the differential operator L defined on functions with domain Ω. That is, we are looking for the smallest real numbers $\Lambda_1 \leqq \Lambda_2 \leqq \cdots \leqq \Lambda_q$ and functions $U_1, U_2, \cdots, U_q$ so that

$$\begin{aligned} &LU_i - \Lambda_i U_i = 0 \quad \text{on } \Omega, \\ &U_i = 0 \quad \text{on } \partial\Omega, \\ &\langle U_i, U_j \rangle = L_{ij}, \quad i, j = 1, 2, \cdots, q. \end{aligned} \tag{4.1}$$

Here we assume that L is an elliptic, self-adjoint differential operator on Ω. (Our work extends naturally to the generalized eigenproblem $LU - \Lambda MU = 0$ with appropriate assumptions on M. We restrict our attention to the case $M = I$ for simplicity.)

Several alternatives can be used in specifying the accuracy required of the eigenvalues and their eigenvectors. For example, a mesh size can be prespecified, determining the discrete operator whose first q eigenvalues approximate those of L. In this case, it is required to find the eigenvalues and eigenvectors to the level of truncation error, that is, to the level of accuracy determined by the extent to which

the discrete eigenvalues and eigenvectors approximate those of the continuous operator. Solving beyond this level does not necessarily improve the error between the approximations and the solutions of the continuous problem. Note in this case that the accuracy is not uniform; that is, the error is larger in the higher eigenvalues than the lower ones since higher eigenvalues generally have larger truncation errors.

Another way to define the accuracy requirements is to specify each eigenvalue error explicitly by giving a fixed tolerance for all eigenvalue approximations. This method is more intricate, since different fine grids will usually be required for different eigenvalue/eigenvector approximations.

The method presented in this paper is designed according to the first accuracy criterion, although the design and testing of an algorithm for use with the second is discussed briefly in § 6. We first examine the case where only one eigenvalue is desired. This avoids this accuracy question yet clarifies some of the basic processes involved in the full algorithm.

Following the notation introduced in § 3, the current G^k approximations to the ith G^l eigenvector and eigenvalue are denoted by u_i^k and λ_i^k, respectively, where G^l is the currently finest grid. Upper case is reserved for exact solutions so that U_i^l and Λ_i^l are the exact G^l solutions. When no ambiguity exists, superscripts and subscripts will be dropped.

5. The method for the first eigenvalue.

5.1. The basic method. The full algorithm is listed in the appendix. In this section we discuss the essentials for a version of this method as it applies to the computation of the first eigenvalue only.

The first step is to obtain the coarsest grid approximation to the first eigenvalue and its eigenvector. Some initial approximation u^1 on this grid is chosen at random, although whatever information is known about the first eigenvector may be used here to provide a better initial guess. λ is chosen as a suitable approximation to the first eigenvalue of the operator. On this coarse grid, relaxation is performed on the equation $L^1u^1-\lambda u^1=0$ with λ fixed, followed by an update of λ by computing the Rayleigh quotient

$$\lambda=\langle L^1u^1, u^1\rangle/\langle u^1, u^1\rangle.$$

This process is repeated until a fairly accurate solution emerges. The vector is then normalized. In the experiments reported in this paper, 15 iterations were generally sufficient, although this value depends on the mesh size of the coarsest grid and the separation of the first and second eigenvalues there (cf. Kahan [6]). In any case, these iterations cost very little since the coarsest grid has only a few points.

Once a solution has been obtained on a level $l-1$, u^{l-1} is cubically interpolated to level l, where it is then improved by one fixed multigrid cycle from level l to level 1 and back. On all but the coarsest grid of this cycle, relaxation is performed holding λ fixed. Two relaxation sweeps per grid ($\nu_1=\nu_2=2$) usually ensure sufficient smoothing and elimination of high frequencies introduced by coarse grid corrections. Thus, the cycle begins with a fixed number of level l relaxation sweeps on the approximate vector followed by FAS transfer of the problem to the next coarser grid. This is repeated until the coarsest grid is reached, where the nonlinear problem resulting there from FAS transfers is solved by repeated relaxation and normalization of the approximate vector and update of λ by (5.6). FAS interpolation (3.3) is then used to transfer the corrected solution back to the next finer grid, where a fixed number of relaxation sweeps is performed, followed by transfer of the approximation to a finer

grid until level l is again reached. Relaxation on u^l is then performed to complete the level l cycle is then increased by 1.

The process of cubic interpolation followed by one multigrid cycle on level l is then continued to yet finer grids until a solution is obtained on level m, the finest grid to be used.

The remainder of this section is devoted to a detailed discussion of certain aspects of this method.

5.2. The coarse grid problem. The coarse grid equation is derived in much the same way as described in § 3. Specifically, let the currently finest grid problem be written as

$$L^l U^l - \Lambda^l U^l = 0, \qquad \eta^l(U^l) = 1, \tag{5.1a,b}$$

where η^l is a normalization constraint functional which specifies the size of the solution. This constraint is necessary for uniqueness since a solution U^l to (5.1a) is specified only up to a multiplicative constant. The specific nature of η will be discussed below. If we have an approximation u^l to U^l, and the error has been smoothed by relaxation, then the G^{l-1} equations are

$$L^{l-1}U^{l-1} - \Lambda^{l-1}U^{l-1} = \tau^{l-1}, \qquad \eta^{l-1}(U^{l-1}) = \sigma^{l-1},$$

where $\tau^{l-1} = L^{l-1}I_l^{l-1}u^l - I_l^{l-1}L^l u^l$ and σ^{l-1} is some appropriate quantity. In general, the G^k problem is

$$L^k U^k - \Lambda^k U^k = \tau^k, \qquad \eta^k(U^k) = \sigma^k \tag{5.2a, b}$$

with $\tau^k = I_{k+1}^k \tau^{k+1} + L^k I_{k+1}^k u^{k+1} - I_{k+1}^k L^{k+1} u^{k+1}$, $\tau^l = 0$.

To see what σ^k should be, it is necessary to examine the goal of the coarse grid solution process. Since Λ^k is unknown, there are fewer equations than unknowns, hence (5.2a) can have an infinite number of solutions. Condition (5.2b) is needed in order to uniquely define the solution. One possible way to define σ^k, analogous to the definition of τ^k, is

$$\sigma^k = \sigma^{k+1} + [\eta^k(I_{k+1}^k u^{k+1}) - \eta^{k+1}(u^{k+1})], \qquad \sigma^l = 1. \tag{5.3}$$

This definition arises from the attempt to force the corrected level l solution to satisfy (5.1b). However, this is not actually necessary, since on the fine grid a solution of any reasonable size is acceptable. Thus, it is acceptable to use a constraint which seeks to maintain the size of the solution, and which, together with (5.2a), provides a zero correction at convergence. Such a constraint is given by

$$\sigma^k = \eta^k(I_{k+1}^k u^{k+1}). \tag{5.4}$$

This constraint is easier to enforce than (5.3) and appears to work well experimentally, so we adopt this definition for our algorithm.

The most obvious choice for the normalization functional is

$$\eta^i(u^i) = \langle u^i, u^i \rangle.$$

However, since the condition

$$\langle u^k, u^k \rangle = \sigma^k$$

when coupled with (5.2a) admits two related solutions, one of which is essentially the negative of the desired one, it is possible to converge to the wrong one on the coarsest grid. In addition to obtaining a bad eigenvalue approximation, this would cause the

corrected approximation to lose most of the eigenvector component that we really want. It is therefore better to use a linear normalization constraint to ensure that the extraneous solution is eliminated. To maintain the "sign" of the coarse grid solution, we can choose $\eta^k = \langle u^k, I_{k+1}^k u^{k+1} \rangle$ so that the normalization constraint then becomes

$$\langle u^k, I_{k+1}^k u^{k+1} \rangle = \langle I_{k+1}^k u^k, I_{k+1}^k u^{k+1} \rangle. \tag{5.5}$$

This normalization constraint, which we adopt here, ensures that the coarse grid solution is approximately in the same direction as the current fine grid approximation, and is compatible at convergence with (5.1a).

Relaxation on the finest few grids affects high-frequency error, which is generally small compared to the solution norm, so the size of the solution changes only slightly. However, on the coarser grids (i.e., for $k = 1$ or 2), low-frequency changes are made and the magnitude of the major component of the eigenvector approximation can change significantly. For this reason, we enforce the normalization constraint (5.5) on the coarse grids only.

For the eigenvalue calculation, note that τ^k naturally provides the correction needed to compute a Rayleigh quotient on the coarser grid. That is, λ can be updated by the relation

$$\lambda = \frac{\langle L^k u^k - \tau^k, u^k \rangle}{\langle u^k, u^k \rangle}. \tag{5.6}$$

5.3. Relaxation. Gauss–Seidel relaxation with a shift parameter for equation (5.2a) was used in the experiments reported in this paper. Let M^k be the lower triangular part of L^k (including the main diagonal), u^k and λ the current approximations to U^k and Λ^k and $\bar{u}^k$ the approximation after relaxation. For each constant shift μ, this relaxation scheme is given by

$$\bar{u}^k = u^k + (M^k - \mu I)^{-1}(\tau^k - (L^k - \lambda I)u^k). \tag{5.7}$$

A useful tool for analyzing (5.7) is the smoothing rate analysis exemplified in [2]. For simplicity, let L^k be the usual five-point approximation to $-\Delta$. If we describe the grid by $G^k = \{(\alpha h_k, \beta h_k) \in \Omega: \alpha, \beta \text{ integers}\}$, where Ω is the function domain for L^k, then a given Fourier error component before and after the relaxation sweep is denoted by $A_\theta e^{i(\theta_1\alpha+\theta_2\beta)}$ and $\bar{A}_\theta e^{i(\theta_1\alpha+\theta_2\beta)}$, respectively. Letting the corresponding component of the actual solution U^k be given by $B_\theta e^{i(\theta_1\alpha+\theta_2\beta)}$, then $\bar{A}_\theta$ is given as a function of A_θ and B_θ by the following equation corresponding to the relaxation scheme (5.7):

$$\begin{aligned}\bar{A}_\theta = &-A_\theta(e^{i\theta_1} + e^{i\theta_2} + h^2(\lambda - \mu))/(e^{-i\theta_1} + e^{-i\theta_2} - 4 + h^2\mu) \\ &+ h^2(\Lambda^k - \lambda)B_\theta/(e^{-i\theta_1} + e^{-i\theta_2} - 4 + h^2\mu).\end{aligned} \tag{5.8}$$

If we define

$$C_\theta = h^2(\Lambda^k - \lambda)B_\theta/(e^{i\theta_1} + e^{i\theta_2} + e^{-i\theta_1} + e^{-i\theta_2} - 4 + h^2\lambda),$$

then we obtain

$$\left|\frac{\bar{A}_\theta - C_\theta}{A_\theta - C_\theta}\right| = |e^{i\theta_1} + e^{i\theta_2} + h^2(\lambda - \mu)|/|e^{-i\theta_1} + e^{-i\theta_2} - 4 + h^2\mu|. \tag{5.9}$$

Note that if $h^2(\lambda - \mu)$ and $h^2\mu$ are small, as is generally the case on finer grids with a proper choice of μ, and if C_θ is small compared to A_θ, then the (θ_1, θ_2) component smoothing rate approaches that for the Poisson problem. On finer grids, $h^2(\Lambda - \lambda)$ is small, even when the eigenvalue approximation has not been updated, and for higher

frequencies, the denominator of C_θ is relatively large. In addition, B_θ is generally small since the solution is smooth and has very small high-frequency components. On coarser grids, however, h^2 can be of significant size, so that C_θ is no longer small compared to A_θ, causing a deterioration of the smoothing rate. To reduce $\Lambda - \lambda$ and thereby minimize the affect of C_θ, we update λ on the coarser grids where the Rayleigh quotient is inexpensive to compute.

The experiments reported in this paper used $\mu = 0$, that is, relaxation without a shift. Generally, this works well if the first eigenvalue is small compared to the spectral radius of L^k, which is always the case here. Relaxation with the shift $\mu = \lambda$ would actually yield slightly better smoothing rates in this case, and for some problems, such as for $-\Delta + cI$ with large c, a nonzero shift may be necessary. However, when a shift is used, care must be taken on coarser grids to avoid letting the denominator of (5.9) become too small since this could lead to substantial magnification of high frequencies. A robust algorithm should incorporate a full shift ($\mu = \lambda$) on finer grids and adjust the shift on coarser grids in order to ensure convergence there. This is especially true for the initial stages of obtaining coarse grid eigenvalue approximations and for computation of higher eigenvalues.

6. The full algorithm.

6.1. General discussion. There are several possibilities for extending the method described in § 5 to apply to the computation of the first q eigenvalues and eigenvectors of L. They differ mostly by the degree to which they handle the eigenvector computations simultaneously. For example, a fully simultaneous extension of the method of § 5 would update all eigenvector and eigenvalue approximations together, both in the fine grid iterations and coarse grid corrections. A fully sequential process would attempt to compute to convergence each eigenvector and eigenvalue in turn. Of course, both methods require additional processes to ensure numerical separateness of the approximations so that they do not all converge to the same eigenvectors. This usually amounts to some sort of orthogonalization process that reflects the orthogonality of the eigenvectors themselves.

One attribute of the full simulataneous method is its ability to use Ritz projections with all of its attendant advantages. For example, as with conventional techniques such as the power method (cf. [8]), the emphasis is placed on producing a good approximation to the subspace spanned by the first q eigenvectors of L. This has many subtle advantages, even more than with the conventional uses of Ritz, but the most direct is that convergence of a specific eigenvalue depends now on its separation not from its neighbor but from the $(q+1)$st eigenvalue of L. The major disadvantage with the fully simultaneous method (and one of the main advantages of the fully sequential one) is storage requirements. All vector approximations must be maintained on all levels of the multigrid cycles. The additional storage is up to $\frac{2}{3}$ of the storage needed (in any case) for storing all of the vectors on the finest grid (since on coarser grids both u^k and τ^k are stored).

The method we propose is intermediate to these two extremes. To retain the advantages of each, we carry all vectors simultaneously through the FMG process by maintaining approximations to all of the eigenvectors and by performing orthogonalization and Ritz projections on the currently finest grid, and we proceed sequentially within the cycling scheme by performing in turn a fixed multigrid cycle on each currently finest grid eigenvector approximation. Sequential use of the multigrid cycling process is apparently no less effective than, yet reduces the storage requirements of, the simultaneous approach. The steps of our algorithm are listed in the appendix and loosely described as follows.

6.2. Initial coarse grid approximations. The first step is to obtain coarse grid approximations to as many of the desired eigenvectors as possible. In general, the *correspondence* between eigenfunctions of the continuous operator and eigenvectors of a discrete approximation to that operator is not exact; that is, the eigenvector corresponding to the ith eigenvalue on some L^h may be a closer approximation to the jth eigenfunction of L with $i \neq j$. This may not necessarily cause problems if i and j are less than q. However, if a coarse grid eigenvector is computed which does not correspond to one of the desired eigenvectors on the finer grids, then it is unlikely that further finer grid work will achieve the desired accuracy. This is less likely to happen if the eigenvalues computed on a coarse grid are limited to a fixed part of the spectrum. Thus, on the coarsest grid G^1, we approximate only $c|G^1|$ eigenvalues and eigenvectors, where $|G^1|$ represents the number of interior G^1 grid points and $c<1$. $c=\frac{1}{4}$ is usually safe, although at times a higher value was used in the experiments reported in this paper since the correspondence for our problems was known to be exact.

The relaxation steps for approximating the eigenvalues and eigenvectors on the coarsest grid are the same as for one eigenvalue with the addition of a Gram–Schmidt orthogonalization step after each sweep. Thus, the ith eigenvector approximation is kept orthogonal to u_j^1, $j=1,\cdots,i-1$, and normalized only at convergence. Once all of the eigenvectors on G^1 have been suitably approximated (i.e., we have accepted $\min\{q, c|G^1|\}$ approximations), then a Ritz projection is performed (see § 6.3). The resulting vectors are then cubically interpolated to the next finer grid, G^2.

In general, once starting vectors are cubically interpolated to level l, one multigrid cycle as described in § 5, along with orthogonalization conditions given in § 6.3 below, is then performed on each u_i^l individually. The coarsest grid $j \geqq 1$ used in this cycle for u_i^l is one on which u_i first appeared (i.e., j is the smallest index so that $i \leqq \min\{q, c|G^j|\}$). If $q > c|G^{l-1}|$, then each of the vectors u_i^l, $c|G^{l-1}| < i \leqq \min\{q, c|G^l|\}$, are computed by the coarse grid process described above (i.e., by relaxation sweeps, each sweep being followed by orthogonalization with respect to $u_1^l,\cdots,u_{i-1}^l$). All the vectors are then cubically interpolated to G^{l+1} and the process is continued until $l=m$.

6.3. Ritz projection. As stated before, once the vectors $u_1^l,\cdots,u_q^l$ have been corrected by multigrid cycles on G^l, the subspace $X=\text{span}\{u_1^l,\cdots,u_q^l\}$ is a good approximation to the subspace spanned by the eigenvectors $U_1^l,\cdots,U_q^l$ that we seek. Ritz projection is a process which finds $\bar{u}_1^l,\cdots,\bar{u}_q^l$ in X and $\bar{\lambda}_1,\cdots,\bar{\lambda}_q$ so that the orthogonal projection of such $L^l\bar{u}_i^l-\bar{\lambda}_i^l\bar{u}_i^l$ onto X is zero. This ensures that any eigenvector of L^l contained in X will be found by Ritz projection. More generally, it will determine a basis for X that is closest to the U_i^l in some sense.

To determine the Ritz vectors we first perform a Gram–Schmidt orthonormalization on $u_1^l,\cdots,u_q^l$, resulting in vectors $\tilde{u}_1^l,\cdots,\tilde{u}_q^l$. Then $\mathcal{U}\mathcal{U}^T$ is an orthogonal projection operator onto X, where

$$\mathcal{U}=[\tilde{u}_1^l,\cdots,\tilde{u}_q^l].$$

Any vector in X can thus be written as $\mathcal{U}z$, where z is a q-vector. Letting $\bar{u}_i^l=\mathcal{U}z_i$, then Ritz projection attempts to find z_i, $\bar{\lambda}_i$, $i=1,2,\cdots,q$, so that

$$\mathcal{U}\mathcal{U}^T(L^l\mathcal{U}z_i-\bar{\lambda}_i\mathcal{U}z_i)=0$$

or, since $\mathcal{U}$ is full rank, so that

$$\mathcal{U}^TL^l\mathcal{U}z_i-\bar{\lambda}_iz_i=0.$$

Hence, the Ritz process requires the solution of a $q \times q$ symmetric eigenvalue problem. Since q is small relative to the size of the fine grid, then the work involved is small. Note that the resulting vectors $\bar{u}_i^l$ form an orthonormal set and that the eigenvalues produced are the Rayleigh quotients of these vectors. This follows clearly from the condition that $L^l\bar{u}_i^l - \lambda_i\bar{u}_i^l$ is orthogonal to X and hence $\bar{u}_i^l$.

Ritz projection could be performed on a coarse grid using FAS approximations to the entries in the Ritz matrix, $\mathcal{U}^T L^k L\mathcal{U}$, but it is better to wait until the coarse grid cycles have been completed on the currently finest grid. At that time, the q-dimensional eigenspace approximation has been corrected. Note that no more fine grid work is needed to perform Ritz on the fine grid than on the coarse grid since the FAS approximation to the Ritz matrix involves calculations on the finest grid. An added advantage of using Ritz on the fine grid after the cycles is that the results are generally more accurate.

6.4. Coarse grid corrections. The fine grid problem (4.1) we seek to solve is equivalent to finding the smallest $\Lambda_1^h \leqq \Lambda_2^h \leqq \cdots \leqq \Lambda_q^h$ and $U_1^h, U_2^h, \cdots, U_q^h$ so that for $i = 1, 2, \cdots, q$,

$$L^h U_i^h - \Lambda_i^h U_i^h = 0, \qquad U_i^h = 0 \quad \text{on } \partial\Omega,$$

$$\langle U_i^h, U_j^h \rangle = \delta_{ij}, \qquad j = 1, 2, \cdots, i.$$

Since the Ritz process requires only a numerically well-determined subspace for the eigenspace from which it computes the eigenvalue approximations, the aim of the multigrid cycle performed on the ith eigenvector approximation is to produce a vector which differs from U_i^h only in the directions of the vectors U_j^h, $j \leqq q$. Thus strict orthogonalization is unnecessary, and we can instead try to maintain the amount of separation that already exists between the vectors. We will refer to these separation constraints as orthonormalization conditions, although it should be understood that these only approximate true orthonormality. The multigrid cycle for the ith vector will tend to produce an approximation to the vector with minimum Rayleigh quotient which satisfies these separation constraints. That is, components of higher eigenvectors are eliminated from the approximation, and those of the previously computed eigenvector approximations remain unchanged. Relaxation changes low-frequency error components only very slightly, and since these components dominate the approximation, then vector norms and the amount of vector separation are approximately maintained during relaxation on the fine grid. Thus normalization and orthogonalization may be reserved for use on coarse grids only, where low-frequency changes will occur.

Orthogonalization is therefore performed only on the very coarsest grids, with the exception that at the termination of each multigrid cycle on the current finest grid G^l, Gram–Schmidt orthonormalization is used to start the Ritz process. Thus, during each cycle, the separation exhibited by the fine grid vectors results only from them being the interpolants of the orthonormal grid $l-1$ approximants. This separation is, however, adequate to provide good numerical determination of the subspace in which they belong. Thus, our FAS orthonormalization conditions are designed to maintain this separation rather than the much more difficult task of maintaining actual orthogonality. The (approximate) orthonormalization constraint, analogous to that for normalization in (5.3), is given by

$$\langle u_j^k, I_{k+1}^k u_i^{k+1} \rangle = \sigma_{ij}^k, \qquad i = 1, \cdots, j, \tag{6.1}$$

where $\sigma_{ij}^k = \langle I_{k+1}^k u_j^{k+1}, I_{k+1}^k u_i^{k+1} \rangle$. Note that the case $i = j$ is the normalization constraint. Then the coarse grid equations are, for $i = 1, 2, \cdots, q$,

$$\begin{aligned} L^k U_i^k - \Lambda_i^l U_i^k &= I^k, \\ \langle U_i^k, I_{k+1}^k U_j^k \rangle &= \sigma_{ij}^k. \end{aligned} \tag{6.2}$$

Relaxation on the coarse grids tends to increase the components in the direction of previous eigenvectors, so for $i < j$, (6.1) is enforced by subtracting a multiple of $I_l^k u_i^l$ from u_j^k, and the equation for $i = j$ is enforced by multiplication by some constant. Since $\sigma_{ij}^k \neq 0$ for $i < j$, unlike true Gram–Schmidt orthogonalization, (6.1) is not exactly satisfied after one pass through the equations. However, experiments show that further work to enforce (6.1) is neither needed nor helpful.

We rely mainly on the coarse grid solution process to eliminate error components of u_i^l in the direction of U_{q+1}^l and other higher low frequencies. Since relaxation on intermediate grids does not significantly affect these frequencies, the approximate action of coarse grid solution on a component U_j is given by

$$I_1^l (L^1 - \Lambda_i^l I)^{-1} (L^1 I_l^1 - I_l^1 L^l) U_j^l.$$

Ignoring grid transfer errors, this means that the U_j^l component in u_i^l is approximately multiplied by γ_j, where

$$\gamma_j = \frac{\Lambda_j^1 - \Lambda_j^l}{\Lambda_j^1 - \Lambda_i^l}.$$

If γ_{q+1} is less than 1 for all $i \leqq q$, then the eigenspace approximation will improve since higher frequencies are reduced at a greater rate. However, this quantity generally depends on q and i and the type of discretization. For programming convenience in the work reported in this paper, 5-point stencil discrete operators were used on all grids. In this case, for a positive definite operator, the discrete eigenvalue approximations generally decrease as the mesh size increases. Thus, for some q, γ can be greater than 1 when coarse grid truncation error exceeds the eigenvalue separation $\Lambda_{q+1}^l - \Lambda_q^l$. For such problems, a variational formulation of the problem may be useful. Here, the coarse grid and grid transfer operators satisfy, up to a scalar factor,

1. $A^{2h} = I_h^{2h} A^h I_{2h}^h$,
2. $I_h^{2h} = I_{2h}^{hT}$.

Using such problem formulation yields discrete eigenvalues which decrease as h decreases. Thus if $\Lambda_i^l < \Lambda_j^l$ then

$$\gamma_j = \frac{\Lambda_j^1 - \Lambda_j^l}{\Lambda_j^1 - \Lambda_i^l} < \frac{\Lambda_j^1 - \Lambda_j^l}{\Lambda_j^1 - \Lambda_j^l} = 1.$$

This avoids any coarse grid anomalies and ensures coarse grid convergence, even when these grids are very coarse.

The coarsest grid used in multigrid cycling, when G is the currently finest grid in the FMG process, should be G^k, where k is $\min\{j: q \leqq c|G^j|\}$. This ensures that relaxation is not performed on levels for which the G^l eigenvectors we seek do not correspond to those on the coarse grid. On too coarse a level, eigenvalue ordering can be different than on G^l, and relaxation there can introduce error in the direction of higher G^l eigenvectors. Another reason not to go to such a coarse grid, even for low eigenvalues, is that most of the correction provided there will be in the direction of eigenvectors belonging to the subspace that we are trying to approximate. More specifically, if $|G^1|$ is not large compared to q, then there is only a negligible change

in the approximate subspace X on G^m due to the corrections computed on G^1. Use of the Ritz process allows us to take this view that we are concerned only with errors in the subspace approximation, not the eigenvectors themselves.

6.5. An alternate accuracy criterion. The results and methods so far have been geared towards finding q eigenvalues of L to within truncation error defined by a prespecified grid mesh size h. Truncation error generally increases with λ, so the same accuracy is not achieved for all eigenvalues. If one wishes to compute the first q eigenvalues of L to within some given tolerance ε, limited experiments indicate that this can be achieved without performing relaxation on the eigenvalues that have already been accepted because they are accurate to that tolerance. These accepted eigenvectors need not even be included in the Ritz process, although they should be carried to the finer grids demanded by the higher eigenvalues and used in the coarse grid and pre-Ritz orthogonalization processes there. Also, if for some reason the discrete eigenvalues are wanted to some level of tolerance smaller than truncation error for a fixed h, we can make the first cycle for all vectors as before, with more work (i.e., further cycles) performed only for higher vectors, using the lower ones again in orthogonalization only.

In both of the above cases, if all vectors are used in the Ritz projection, the convergence rates for the higher eigenvectors are close to those obtained by the more expensive algorithm that includes the lower eigenvalues in all processes. In the second case, the rates are almost identical.

7. Work and storage requirements. Since normalization, orthogonalization and computation of the orthonormalization correction terms and the Rayleigh quotient are performed primarily on coarser grids, we neglect these processes in developing the following overall work estimates. For convenience, let $n_k = |G^k|$ and assume $n_k = \frac{1}{4} n_{k+1}$ for $k = 1, 2, \cdots, m-1$. Let αn_k represent the number of operations in one G^k relaxation sweep. This is essentially the cost of one matrix multiply. α is approximately twice the number of nonzero entries per row in L^l. (In some cases, such as a uniform discretization of the Laplacian, α is much smaller than this.) An inner product on G^k is assumed to cost $2n_k$ operations. The amount of work involved at one stage of the FMG process results from the following computations:

1. Cubic interpolation of u_i^{l-1} to u_i^l for $i = 1, \cdots, q$.
2. $\nu_1 + \nu_2$ relaxation sweeps on u_i^k, $i = 1, \cdots, q$, $k = 1, \cdots, l$.
3. Coarse grid τ_i^k calculation for $k = 1, \cdots, l-1$, $i = 1, \cdots, q$.
4. FAS interpolation of u_i^{k-1} to u_i^k, $i = 1, \cdots, q$, $k = 2, \cdots, l$.
5. Ritz projection on G^l, which includes:
   a. orthonormalization of $u_1^l, \cdots, u_q^l$;
   b. compilation of the Ritz matrix $[u_i^{l\,T} L^l u_j^l]_{i,j}$, $i, j = 1, \cdots, q$;
   c. solving for all the eigenvectors and eigenvalues of the Ritz matrix;
   d. computing the new vectors $\bar{u}_1^l, \bar{u}_2^l, \cdots, \bar{u}_q^l$.

The work involved in each of these processes is measured as follows:

1. $\frac{15}{4} n_l \cdot q$.
2. $\frac{4}{3}\alpha(\nu_1 + \nu_2) n_l \cdot q$.
3. $\frac{8}{3}(\alpha+1) n_{l-1} q = \frac{2}{3}(\alpha+1) n_l \cdot q$.
4. $\frac{11}{4} n_l \frac{4}{3} \cdot q = \frac{11}{3} n_l \cdot q$.
5. a. $(2q^2 + q) n_l$;
   b. $\alpha q \cdot n_l + q(q+1) n_l$;
   c. $O(q^3)$;
   d. $q(2q-1) n_l$.

This gives a total G^l operation count, W_l, of

$$W_l = \tfrac{4}{3}(\nu_1 + \nu_2 + \tfrac{5}{4})q\alpha n_l + [\tfrac{109}{12}q + 5q^2]n_l + O(q^3).$$

The total operation count, $W = \sum_{l=1}^{m} W_l$, for the complete FMG algorithm is therefore

$$W \approx \tfrac{16}{9}(\nu_1 + \nu_2 + \tfrac{5}{4})q\alpha n_m + [12q + \tfrac{20}{3}q^2]n_m + mO(q^3).$$

Note that m should be proportional to $\log n_m$ so that

$$W = n_m O(q^2) + \log n_m O(q^3).$$

Storage is needed for all vectors on all grids, although future work on small-storage algorithms (see [2, § 7.5]) may drastically reduce that requirement. Since coarse grid problems are inhomogeneous, storage is also required for the right-hand side of these problems. However, the sequential manner in which the coarse grid cycles are performed means that only one vector per grid is needed to store the right-hand side τ^k. The finest grid actually needs no storage for this, since the problem is homogeneous. All other storage is at most proportional to the product of the number of points in one direction on the fine grid and the number of eigenvalues being computed. Thus, disregarding locations needed for bookkeeping, the total storage location requirements are approximately

$$S = \tfrac{4}{3}qn_m + \tfrac{1}{3}n_m.$$

8. Computational results. The model problem used in our experiments is given by

$$\begin{aligned} -\Delta u + 10y(\sin 3\pi x)u &= \lambda u \quad \text{on } \Omega = [0,1]\times[0,1], \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{8.1}$$

The Laplace operator by itself has several properties not typical of more general operators which made it unsuitable for reliable tests. The term added to the Laplacian in (8.1) causes the multiple eigenvalues of the Laplace operator to be perturbed, yielding instead sets of close eigenvalues. This poses more of a challenge to our algorithm. Moreover, the eigenvectors are altered so that they are not exactly represented by the discrete problem as they are for the Laplacian alone.

Unless otherwise indicated, $h_1 = \frac{1}{4}$ and $m = 4$ so that $h_m = \frac{1}{32}$. Also, $\nu_1 = \nu_2 = 2$.

A closed form solution for the eigenvalues and eigenvalue discretization error of this problem is not known, but can be closely approximated by carrying out the test results farther than the number of cycles and the fineness of h_m than we report. The "exact" values Λ_n^0 used in Table 1 were calculated by extrapolating the computed solutions on levels $m = 4$ and $m = 5$. h in Table 1 represents $h = h_4 = \frac{1}{32}$.

TABLE 1

n	Λ_n^0	Λ_n^h	$\lvert\Lambda_n^0 - \Lambda_n^h\rvert$	$\lVert L^h U_n^0 - \Lambda_n^0 U_n^0\rVert$
1	18.73558161	18.71847149	.0171	.0407
2	48.32534796	48.18927363	.136	.142
3	51.69556290	51.56004355	.136	.148
4	81.32645700	81.07201016	.254	.263
5	97.65037417	97.00117915	.649	.651
6	100.2221931	99.57484220	.647	.654
7	129.8746755	129.1084354	.766	.773
8	130.6674040	129.8996943	.768	.771
9	166.65623	164.6376509	2.02	2.02
10	169.0329	167.0085449	2.02	2.03

The last column in Table 1 is the norm of the residual of the solution to the continuous problem projected onto the $h = \frac{1}{32}$ grid. This is also an approximation, but is close enough to serve as a quantity against which to measure convergence of the discrete solution. That is, when the grid h residual is comparable to the residual formed by the projected continuous solution, then we conclude that the discrete error is the same order as the truncation error. The algebraic error in the eigenvalue is not necessarily a good measure of the algebraic error in the eigenvector, the former being approximately proportional to the square of the latter.

The method for one eigenvalue and $h = \frac{1}{32}$ is given in Table 2. The amount of work performed in relaxation, which dominates the overall work in this case, is equivalent to about seven sweeps on the fine grid.

TABLE 2

λ_1^h	$\lvert\lambda_1^h - \Lambda_1^h\rvert$	$\lVert L^h u_1^h - \Lambda_1^h u_1^h\rVert$
18.71871030	2.39×10^{-4}	1.40×10^{-2}

The results for ten eigenvalues and $h = \frac{1}{32}$ are shown in Table 3. A comparison with Table 1 shows that the problem is solved to below the level of truncation error. Note that the accuracy is better relative to this truncation error for the lower eigenvalues. If more eigenvectors are included in the process (whether or not they converge), all approximations are improved. When 15 eigenvectors were included in a test that we ran, although the last two failed to converge, the residual norm of the 10th eigenvector decreased by about a factor of 2. This is not of too much practical importance, however, since even though it suggests that less work is required for the lower eigenvalues when Ritz projection is used, the error decrease is actually due mainly to elimination of low-frequency error.

TABLE 3

n	λ_n^h	$\lvert\lambda_n^h - \Lambda_n^h\rvert$	$\lVert L^h u_n^h - \lambda_n^h u_n^h\rVert$
1	18.71847153	3.40×10^{-8}	4.26×10^{-3}
2	48.18927456	9.31×10^{-7}	2.04×10^{-2}
3	51.56004444	8.90×10^{-7}	2.32×10^{-2}
4	81.07201416	4.00×10^{-6}	3.80×10^{-2}
5	97.00123840	5.93×10^{-5}	1.64×10^{-1}
6	99.57489151	4.93×10^{-5}	1.56×10^{-1}
7	129.1088552	4.20×10^{-4}	2.64×10^{-1}
8	129.9001827	4.88×10^{-4}	2.77×10^{-1}
9	164.6602169	2.26×10^{-2}	1.74
10	167.0701555	6.16×10^{-2}	1.72

As explained in § 6, low-frequency error can arise if the coarse grid solution does not provide a good correction to the approximations on finer grids. In these cases a cycle with more visits to coarser grids may help. Table 4 gives the results for the so-called "W-cycle" for ten eigenvalues. A W-cycle means that the coarse grid correction to any grid k is calculated by *two* cycles on grid $k-1$. The amount of work performed in relaxation is $\frac{3}{2}$ times the amount needed for usual multigrid cycles. The total amount of work is then

$$W \approx \tfrac{8}{9}(3(\nu_1+\nu_2)+\tfrac{5}{2})q\alpha n_m + [12q + \tfrac{20}{3}q^2]n_m + mO(q^3).$$

TABLE 4

n	$\lvert\Lambda_n^h-\lambda_n^h\rvert$	$\lVert L^h u_n^h-\lambda_n^h u_n^h\rVert$
1	3.60×10^{-8}	4.21×10^{-3}
2	6.57×10^{-7}	1.83×10^{-2}
3	6.55×10^{-7}	2.12×10^{-2}
4	2.34×10^{-6}	3.10×10^{-2}
5	3.97×10^{-5}	1.47×10^{-1}
6	3.27×10^{-5}	1.40×10^{-1}
7	4.68×10^{-5}	1.53×10^{-1}
8	5.29×10^{-5}	1.58×10^{-1}
9	2.36×10^{-3}	7.15×10^{-1}
10	1.42×10^{-2}	8.83×10^{-1}

Tables 5 and 6 show the results obtained with red-black ordering in place of lexicographic Gauss–Seidel with usual cycling and W-cycling, respectively. This type of ordering is a more effective smoother than lexicographic, as evidenced by the higher accuracy obtained in the eigenvalues.

TABLE 5

n	$\lvert\lambda_n^h-\Lambda_n^h\rvert$	$\lVert L^h u_n^h-\lambda_n^h u_n^h\rVert$
1	1.23×10^{-8}	2.98×10^{-3}
2	8.40×10^{-8}	2.39×10^{-2}
3	1.03×10^{-7}	2.33×10^{-2}
4	5.54×10^{-7}	6.50×10^{-2}
5	5.64×10^{-6}	1.89×10^{-1}
6	8.51×10^{-6}	2.57×10^{-1}
7	1.44×10^{-5}	3.32×10^{-1}
8	2.72×10^{-5}	2.88×10^{-1}
9	9.84×10^{-3}	1.05
10	1.16×10^{-1}	1.96

TABLE 6

n	$\lvert\lambda_n^h-\Lambda_n^h\rvert$	$\lVert L^h u_n^h-\lambda_n^h u_n^h\rVert$
1	4.45×10^{-9}	1.65×10^{-3}
2	2.96×10^{-8}	1.35×10^{-2}
3	3.31×10^{-8}	7.48×10^{-3}
4	7.64×10^{-8}	2.08×10^{-2}
5	9.79×10^{-7}	6.19×10^{-2}
6	1.17×10^{-5}	3.00×10^{-1}
7	1.31×10^{-5}	3.18×10^{-1}
8	1.77×10^{-6}	9.22×10^{-2}
9	2.60×10^{-3}	4.40×10^{-1}
10	3.87×10^{-2}	1.31

The nature of the FMG process is ideal for the use of extrapolation on the eigenvalues approximations. Table 7 shows the eigenvalue approximations extrapolated from the $\frac{1}{16}$ and $\frac{1}{32}$ grids and the accuracy obtained.

TABLE 7

n	$(4\lambda_n^h - \lambda_h^{2h})/3$	$\lvert\Lambda_n^0 - (4\lambda_n^h - \lambda_n^{2h})/3\rvert$
1	18.73554270	3.89×10^{-5}
2	48.32469250	6.55×10^{-4}
3	51.69476513	7.98×10^{-4}
4	81.32502517	1.43×10^{-3}
5	97.64234667	8.03×10^{-3}
6	100.2140080	8.19×10^{-3}
7	129.8582268	1.64×10^{-2}
8	130.6508394	1.66×10^{-2}
9	166.4985283	1.58×10^{-1}
10	168.8655123	1.67×10^{-1}

Appendix. The algorithm is broken into two parts: the FMG Ritz procedure and the CYCLE procedure. CYCLE is called from FMG Ritz. The parameters of the algorithm are as follows:

q: The number of eigenvalues and eigenvectors desired.

m: The number of grids to be used.

ν_0: The number of iterations used to obtain a first approximation to each vector on the coarsest grid.

ν_1: The number of relaxation sweeps performed before transferring the problem to a coarser grid.

ν_2: The number of sweeps performed after the coarse grid correction.

FMG RITZ.

1. Set $n \leftarrow 1$, n max $\leftarrow 1$, $l \leftarrow 1$, $\lambda_0 \leftarrow 0$
2. Set $u_n^l \leftarrow$ Random function $\lambda_n \leftarrow \lambda_{n-1}$
   For $i = 1, 2, \cdots, \nu_0$, do
   $u_n^l \leftarrow$ Relax $(L^l u_n^l - \lambda_n u_n^l = 0)$
   For $j = 1, 2, \cdots, n-1$, do
   $u_n^l \leftarrow u_n^l - \langle u_n^l, u_j^l\rangle u_j^l$
   Set $\lambda_n \leftarrow \langle Lu_n^l, u_n^l\rangle / \langle u_n^l, u_n^l\rangle$
   Set $u_n^l \leftarrow u_n^l / \langle u_n^l, u_n^l\rangle$.
3. If $n = q$, set n max $= q$, k min $= l$,
   Ritz on u_i^l, $i = 1, 2, \cdots, n$ max & go to 4
   If $n < c|G^1|$, set $n \leftarrow n+1$ & go to 2.
   Set n max $\leftarrow n$, k min $\leftarrow l$ & go to 4.
4. If $l = m$, stop.
   Set $l \leftarrow l+1$
   For $i = 1, 2, \cdots, n$ max, do
   $u_i^l \leftarrow I_{l-1}^l u_i^{l-1}$
   cycle $(i, k$ min, $l)$
   Ritz on u_i^l, $i = 1, 2, \cdots, n$ max.
5. If n max $< q$, set $n \leftarrow n$ max $+1$ & go to 2.
   Otherwise go to 4.

CYCLE $(n, k$ min, $l)$.

1. Set $k \leftarrow l$, $\tau_n^k \leftarrow 0$
2. For $i = 1, 2, \cdots, \nu_1$, do
   $u_n^k \leftarrow$ Relax $(L^k u_n^k - \lambda_n u_n^k = \tau_n^k)$
   If $k \leqq k$ min

For $j = 1, 2, \cdots, n-1$, do

$u_n^k \leftarrow u_n^k - (\langle u_n^k, I_l^k u_j^l\rangle - \langle I_{k+1}^k u_n^{k+1}, I_l^k u_j^l\rangle)/\langle I_l^k u_j^l, I_l^k u_j^l\rangle \cdot I_l^k u_j^l$

Set $u_n^k \leftarrow u_n^k \langle I_{k+1}^k u_n^{k+1}, I_l^k u_n^l\rangle/\langle u_n^k, I_l^k u_n^l\rangle$

If $k = k$ min, set $\lambda_n \leftarrow \langle L^k u_n^k - \tau_n^k, u_n^k\rangle/\langle u_n^k, u_n^k\rangle$.

3. If $k = k$ min, go to 5.
4. Set $k \leftarrow k-1$

$\tau_n^k = \tau_n^{k+1} + L^k I_{k+1}^k u^{k+1} - I_{k+1}^k L^{k+1} u^{k+1}$

$u_n^k \leftarrow I_{k+1}^k u_n^k$

Go to 2.

5. For $i = 1, 2, \cdots, \nu_2$, do

$u_n^k \leftarrow \text{Relax}\,(L^k u_n^k - \lambda_n u_n^k = \tau_n^k)$

If $k \leqq k_1$

For $j = 1, 2, \cdots, n-1$, do

$u_n^k \leftarrow u_n^k - (\langle u_n^k, I_l^k u_j^l\rangle - \langle I_{k+1}^k u_n^{k+1}, I_l^k u_j^l\rangle)/\langle I_l^k u_j^l, I_l^k u_j^l\rangle \cdot I_l^k u_j^l$

Set $u_n^k \leftarrow u_n^k \langle I_{k+1}^k u_n^{k+1}, I_l^k u_n^l\rangle/\langle u_n^k, I_l^k u_n^l\rangle \cdot I_l^k u_j^l$

If $k = k$ min, set $\lambda_n \leftarrow \langle L^k u_n^k - \tau^k, u_n^k\rangle/\langle u_n^k, u_n^k\rangle$.

6. If $k = l$, stop.
7. Set $k \leftarrow k+1$

$u_n^k \leftarrow u_n^k + I_{k-1}^k (u_n^{k-1} - I_k^{k-1} u_n^k)$

Go to 5.

REFERENCES

[1] R. G. ALCOUFFE, A. BRANDT, J. E. DENDY, JR. AND J. W. PAINTER, *The multi-grid methods for the diffusion equation with strongly discontinuous coefficients*, Report LA-UR 80-1463, Los Alamos Scientific Laboratory, Los Alamos, NM, 1980.

[2] A. BRANDT, *Multi-level adaptive solutions to boundary value problems*, ICASE Report RC 76-27 1976, Hampton, VA, NASA Langley Research Center; Math. Comp., 31 (1977), pp. 336–390.

[3] A. Brandt, *Guide to multigrid development*, in Multigrid Methods, proceedings of a conference (Köln-Porz, No., 1981), W. Hackbusch and U. Trattenberg, eds. Springer-Verlag, New York, 1982.

[4] D. S. CLEMM AND K. G. GUDERLEY, *Eigenvalue and near eigenvalue problems solved by Brandt's multigrid method*, Univ. of Dayton Research Institute Interim Report AFFDL-TR-79-3147, Dayton, OH, 1979.

[5] W. HACKBUSCH, *Multigrid solutions to linear and nonlinear eigenvalue problems for integral and differential equations*, Report 80-3, Math. Inst. Univ. Köln, 1980.

[6] W. KAHAN, *Relaxation methods for an eigenproblem*, Stanford Technical Report AD-638818, Stanford Univ., Stanford, CA, 1966.

[7] S. F. MCCORMICK, *A mesh refinement method for* $Ax = \lambda Bx$, Math. Comp., 36 (1981), pp. 485–498.

[8] S. F. MCCORMICK AND T. NOE, *Simultaneous iteration for the matrix eigenvalue problem*, J. Lin. Alg. Appl., 16 (1977), pp. 43–56.

[9] L. STRAKHOVSKAYA, *An iterative method for evaluating the first eigenvalue of an elliptic operator*, USSR Comput. Math. and Math. Phys., 17 (1978), pp. 88–101.