A Mesh Refinement Method for $Ax = \lambda Bx$

By Stephen F. McCormick*

Abstract. The aim of this paper is to introduce a simple but efficient mesh refinement strategy for use with inverse iteration for finding one or a few solutions of an ordinary or partial differential eigenproblem of the form $Ax = \lambda Bx$. The focus is upon the case where $A$ and $B$ are symmetric and $B$ is positive definite, although the approaches have a very broad application. A discussion of the combined use of mesh refinement and a correction scheme multigrid technique is also provided. The methods are illustrated by numerical results from experiments with two-point boundary value problems.

1. Introduction. There are presently several areas of activity surrounding the development of numerical techniques for solving the generalized eigenproblem

$Ax = \lambda Bx,$

where $A$ and $B$ are $n \times n$ symmetric matrices and $B$ is positive definite. A brief account of part of this history is given in [11], with more recent results reported in [7] and [10]. See also [9] for the case $B = I$. Of special interest is the large sparse eigenproblem where $n$ is very large and only a small percentage of the entries of $A$ and $B$ are nonzero. Of further interest are problems of this type where the pattern of the nonzero entries in both $A$ and $B$ prohibits direct triangular factorization. For such problems, linear equations involving $A$ and $B$ must generally be solved indirectly, that is, by iteration.

Most problems of this latter type result from discretizing an infinite-dimensional eigenproblem, most notably from that of a partial differential operator. This is normally a major source of numerical difficulty for methods that attempt to solve (1) since it almost guarantees that the relative separation of the eigenvalues is pathologically small. Thus, even though only one or a few of the extreme eigenvalues of (1) are usually needed in such cases, “factorization-free” iterative methods will often fail to provide a good approximation without performing an intolerable amount of computational work; cf., [4]. The limitation of these methods is to be expected, however, since they rely on model assumptions on (1) that do not take much account of the problem origin. In fact, few practical methods presently in use do. (See, however, [6] and [13].) Yet, taking this origin into account can lead to techniques that are remarkably efficient. A scheme of this type is the subject of this paper.

The essence of the method described in the following sections is a combination of the techniques of mesh refinement, inverse iteration, and multigrid. On each grid in the progressive mesh refinement process, an outer loop inverse iteration equation

Received July 16, 1979; revised March 13, 1980.


*This work was supported by the Air Force Office of Scientific Research under AFOSR Grant #76-3019 and the National Science Foundation under grant #MCS78-03847.
is posed which has the effect of linearizing the eigenvalue problem. This allows for
the use of an inner loop correction scheme version of multigrid iteration [5], for
example. Dramatic steps from coarse to finer grids are made with the approximate
eigenvector(s) and corresponding eigenvalue(s) "passed" to the finer grid to act as
an initial guess for one outer loop iteration. The process is terminated when the
truncation error on the present finest grid is smaller than a prescribed tolerance.

The mesh refinement process follows directly from earlier work (i.e., [1], [2], [8])
with Newton's method applied to nonlinear two-point boundary value problems. In
fact, the eigenvalue problem can be recast as a nonlinear problem for which
Newton's method reduces with some modification to RQI, the Rayleigh quotient
iteration (cf., [11]). Some of this earlier work has its parallel in application to (1),
although the present work is distinguished, as we shall see, by a difficulty with the
Rayleigh quotient and an advantage of cubic convergence of RQI.

A closely related approach, which has been considered quite generally in [6] and
is presently being investigated from a practical standpoint by A. Brandt and others,
is to apply FAS, the full approximation multigrid scheme (cf., [5]), directly to the
eigenproblem by treating it as a nonlinear operator equation. Quite roughly
speaking, this can be viewed as an application of an iterative method that has been
linearized at each step for computability. Thus, the roles of inner and outer loop
iteration are just the reverse of the approach of this paper, although the multigrid
involvement somewhat confuses this interpretation. In any event, the FAS ap-
proach is more complex and, at least for well-behaved problems, is probably a little
less efficient. It is also, probably, somewhat harder to direct to the proper
eigenvalue and is less amenable to modifications such as simultaneous iteration;
 cf., [7]. On the other hand, for problems that may benefit from adaptive refine-
ment, the FAS approach should prove to be more versatile.

After the preliminaries of Section 2, Section 3 contains a discussion of the mesh
refinement process, which can be viewed as a scheme for defining RQI by moving
progressively from coarse to finer grids. The suitability for solving the RQI defining
equation by the correction scheme multigrid method, which is a grid cycling
scheme, is discussed briefly in Section 4. In Section 5 we present a theoretical basis
for a variational formulation of the mesh refinement. The last section illustrates the
mesh refinement scheme as it applies numerically to two-point boundary value
problems.

2. Assumptions and Notations. We assume in this and the next section that \( \partial \) and
\( \mathcal{B} \) are mappings from some suitable function space \( \mathcal{C}_1(\Omega) \) into another \( \mathcal{C}_2(\Omega) \), with
\( \mathcal{C}_1(\Omega) \subset \mathcal{C}_2(\Omega) \subset \mathcal{L}_2(\Omega) \). The function domain is \( \Omega \subset \mathbb{R}^d \), where \( d \) is a positive
integer. For each \( h \) in a nonempty indexing set \( \mathcal{H} \subset \mathbb{R}^+ \), with \( n = n(h) \), we
assume the existence of \( n \times n \) matrices \( hA \) and \( hB \) that, together with \( \partial \) and \( \mathcal{B} \),
satisfy the following:

A1. The spectrum of
\[
(2) \quad \partial x = \lambda \mathcal{B} x, \quad x \in \mathcal{C}_1(\Omega) - \{0\},
\]
consists of a discrete set of eigenvalues ordered, counting multiplicities, according to

\[
(3) \quad \lambda_1 < \lambda_2 < \lambda_3 < \lambda_4 < \ldots \to \infty.
\]
(The case where \( \lambda_1 \) is multiple is discussed briefly in Section 3.)
A2. For each $h$ in $\mathcal{K}$, $A_h$ and $B_h$ are symmetric and $B_h$ is positive definite. The eigenvalues of

$$(4_h) \quad A_h x = \lambda_h B_h x, \quad x \in \mathbb{R}^n - \{0\},$$

are denoted by

$$(5_h) \quad \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$ 

For compatibility, on $\mathbb{R}^n$ let the $B$-inner product $\langle x, y \rangle_h$ and induced $B$-norm $\|x\|_h$ denote suitable discrete approximations to the $B$-norm and $B$-inner product in $L_2(\Omega)$. (For example, if $\Omega = \prod_{i=1}^d [0, 1]$ and $h = n^{-1/d}$, we may choose $\langle x, y \rangle_h = h^d x^T B_h y$.) With this inner product, a $B$-orthonormal set of eigenvectors for $(4_h)$ is denoted by

$$(6_h) \quad \{ u_1, u_2, \ldots, u_n \}.$$ 

A3. There exist $c_1, c_2, H \in \mathbb{R}$, and a positive integer $m$ such that

$$(7) \quad |\lambda_1 - \lambda_{1,h}| < c_1 h^m \quad \text{and} \quad |\lambda_2 - \lambda_{2,h}| < c_2 h^m$$

for all $h < H$, $h \in \mathcal{K}$.

A4. Let $h_1, h_2 \in \mathcal{K}$ and write $n_1 = n(h_1)$ and $n_2 = n(h_2)$. Denote by $I_{h_1}^{h_2}$ a mapping from $\mathbb{R}^{n_1}$ into $\mathbb{R}^{n_2}$ that preserves eigenvector approximations for $(4_h)$ and $(4_h')$. Specifically, we assume that for all $\varepsilon > 0$ there exists an $H' = H'(\varepsilon)$ so that, with any $x \in \mathbb{R}^{n_1} - \{0\}$ such that $\|x\|_{h_1} = 1$ and with $y = I_{h_1}^{h_2}(x)/\|I_{h_1}^{h_2}(x)\|_{h_2}$, we have

$$(8) \quad \|y - h_2 u_i\|_{h_2} \leq \|x - h_1 u_i\|_{h_1} + \varepsilon,$$

whenever $h_2 < h_1 < H'$, $h_1, h_2 \in \mathcal{K}$. [Note that when $(4_h)$ represents a variational discretization of (2) as in Section 5, then a natural embedding exists of each $\mathbb{R}^n$ into $L_2(\Omega)$, for which $I_{h_1}^{h_2}$ is essentially the identity and (8) is satisfied by setting $\varepsilon$ equal to the sum of the eigenvector truncation errors on grids $h_1$ and $h_2$. For finite difference discretizations, (8) must also account for the interpolation error incurred in $I_{h_1}^{h_2}$. Since this can be made negligible when $I_{h_1}^{h_2}$ is chosen properly, we can roughly conclude that $\|y - h_2 u_i\|_{h_2} \sim \|x - h_1 u_i\|_{h_1}$]

There are many examples of large, sparse eigenproblems that arise from a setting satisfying A1–A4 and where factorization of either $A$ or $B$ is undesirable. Often they are posed as partial differential eigenproblems such as those that arise in structural analysis (cf. [7]) and diffusion reaction modeling (cf. [4] and [13]). Note that all of the examples in Section 6 satisfy A1–A4.

3. Mesh Refinement. It is natural to attempt to find the smallest eigenvalue of (2) by choosing a grid sequence $h_1 > h_2 > \cdots > h_p$ in $\mathcal{K}$ and performing some sort of iteration on each grid $h_i$ initialized with $I_{h_{i-1}}^{h_i}$ applied to the accepted solution from grid $h_{i-1}$, $i = 2, 3, \ldots, p$. However, this would be worth doing only in connection with an iterative process that converges at least superlinearly since the
essential difficulty with such methods is in obtaining good starting guesses. It is therefore natural to consider mesh refinement in connection with RQI which, ignoring the normalization step, is given by

$$ (hA - R(x^{(k)})hB)x^{(k+1)} = hBx^{(k)}, \quad k = 0, 1, 2, \ldots, $$

where the Rayleigh quotient $R(x) = \langle x, hAx \rangle / \langle x, hBx \rangle$. (We postpone the discussion of how the linear equation in (9) is to be solved until Section 4.) This will usually fail to work because of the nature of $R(x)$, that is, even though an approximate eigenvector on a given grid may be fairly accurate, the Rayleigh quotient is biased well upward by the large eigenvalues present in (4A). The result is that, even for very modest jumps to finer grids, the approximation by (9) will usually shift from the smallest eigenvector to successively higher ones as the refinement progresses. The key in avoiding this unacceptable behavior is to pass from grid $h_i$ to grid $h_{i+1}$ not only the approximate eigenvector but the approximate eigenvalue as well. Except for this simple but critically important step, the mesh refinement strategy is similar to the approach in [8]. We therefore omit any background except to state that the bases for the refinement process include:

B1. for each $i$, compute an approximate smallest eigenvalue on grid $h_i$ to within the truncation error $c_1h_1^m$ of $h\lambda_1$;

B2. attempt to design the refinement so that $h_i/h_{i-1}$ is as small as possible (starting with $i = p$) yet insures that only one iteration in (9) is needed for each $i$; and

B3. determine $h_p$ so that the accepted approximation to $h\lambda_1$ is within a prescribed tolerance ($tol > 0$) of $\lambda_1$.

Simply stated, the preprocessing phase of the mesh refinement procedure begins by choosing initial coarse grids $h_1$ and $h_2 = h_1/2$ and examining the eigenvalues $h\lambda_i, i, j = 1, 2$, in order to: approximate the truncation error expressions in (7); determine $p$ and the final grid size $h_p$ so that

$$ c_1h_p^m < tol $$

(as dictated by B1 and B3 above); choose as large an $h_{p-1}$ as is possible so that $h_{p-1}/h_p$ is a power of 2 and so that the Rayleigh quotient of the first RQI iterate is within $c_1h_p^m$ of $h\lambda_1$ (where RQI is initialized with the first grid $h_{p-1}$ approximation interpolated to grid $h_p$); determine $h_{p-2}$ in a similar manner with respect to $h_{p-1}$; and proceed until $h_2$ satisfies these criteria with respect to $h_3$. The execution phase of the mesh refinement procedure then begins by interpolating the grid $h_2$ approximation (to $h_2u_1$) up to grid $h_3$, to initialize one RQI, and continues by repeating this process until the grid $h_p$ approximation (to $h_pu_1$) is computed via one RQI.

There are many possible and advisable steps that should be taken in a fully implemented procedure that insure that the approximation is to the extreme eigenvalue, that the truncation error estimate is accurate, and that the final approximation is suitable, for example. We comment briefly on a few of them as they arise in the following description of the mesh refinement procedure ($m > 1$, $\gamma > 0$, and $tol > 0$ are assumed to be given):
Step 1. Choose \( h_1 \in \mathcal{R} \) and let \( h_2 = 2h_1 \). Compute (to machine precision) \( h_1 \lambda_j \) and \( h_2 u_j \) and define the approximations to the truncation error coefficients in (7) by
\[
\tilde{c}_j = \gamma |h_1 \lambda_j - h_2 \lambda_j| h_1^{-m}, \quad i, j = 1, 2.
\]

Comment 1. The quantity \( \gamma > 0 \) may be considered as a fudge factor to insure that \( \tilde{c}_j > c_j \) in (7). Note that for “monotonic” discretizations (for which, by definition, \( \lambda_j \) converges monotonically to \( \lambda_j \) as \( h \to 0^+ \)) we should choose \( \gamma > 2m/(2m - 1) \).

The choice for \( h_1 \) should be made based upon some heuristics or by some specific knowledge of the problem (e.g., the number of zeros of \( u_1 \)). With the constraint that \( h_1 u_1 \) is a sensible approximation to \( u_1 \) in this way, \( h_1 \) should be chosen so large as to guarantee that the computational effort expended in Step 1 is small compared to subsequent steps. (Note that it may be necessary to compute all of the eigenvalues of (4A) and (4A) to safely select the approximations for the two smallest. This will mandate a fairly large \( h_1 \).)

For Step 1, we have assumed \( m \) is known, but \( c_1, c_2 \) are not. If \( c_1, c_2 \) are known, the computation on grid \( h_2 \) may be ignored. If \( m \) is known, a third grid \( h_3 \) may be used to “fit” the parameters \( m, c_1, c_2 \) in (7) to the “data” \( h_1 \lambda_j, i = 1, 2, 3, j = 1, 2 \).

Step 2. Perform a test to insure that the (computed) \( h_1 \lambda_1 \) is close enough to \( \lambda_1 \) to act as a shift for RQI, namely, that
\[
\tilde{c}_1 h_2^m < \frac{\lambda_2 - \lambda_1}{2}.
\]

If \( h_2 \) is too large according to the test, reduce \( h_1 \) and repeat, beginning with Step 1. 

Comment 2. A simple general test is
\[
(3\tilde{c}_1 + c_2) h_2^m < h_1 \lambda_2 - h_1 \lambda_1,
\]
although knowledge of the problem setting (e.g., monotonicity as defined in Comment 1) may suggest a somewhat less stringent test.

Step 3. Let \( p = 3 \) and determine the final mesh size \( h_3 \) as the largest element of \( \mathcal{R} \) satisfying
\[
2\tilde{c}_1 h_3^m \leq \text{tol}.
\]
(If \( h_3 > h_2 \), output \( h_2 \lambda_1 \) and \( h_2 u_1 \) and stop.)

Comment 3. The final value of \( p \) is the number of grids determined in the refinement process. The determination of \( h_p \) insures that an acceptable (see B1) approximation to \( h_p \lambda_1 \) is within tol of \( \lambda_1 \).

Step 4. Letting \( i = 2 \) and \( q = c_i^2/m(h_i \lambda_j - h_i \lambda_i)^{-2/m} \), determine whether or not
\[
q h_i^3 \leq h_{i+1}.
\]

If so, go to Step 6. Otherwise, go to Step 5.

Comment 4. We have made a few simplifying but reasonable assumptions here. For example, we assume that eigenvector truncation error is not significantly changed by interpolation. Moreover, we assume that \( \tilde{c}_1, \tilde{c}_2 \) bound \( c_1, c_2 \) and that the eigenvalue truncation error on grid \( h_i, i > 2 \), is negligible compared to \( \lambda_2 - \lambda_1 \). Fortunately, performance of the mesh refinement procedure is very unsensitive to such assumptions and, only for badly behaved problems, need they be reconsidered.
Step 5. Rename each \( h_j, j = 3, 4, \ldots, p \), by calling it \( h_{j+1} \) (\( h_3 \) becomes \( h_4 \), \( h_4 \) becomes \( h_5 \), etc.) and increment \( p \) by one. Now, determine as large an \( h_3 \) as is possible so that \( h_3/h_4 \) is a power of 2 and so that \((10)\) is satisfied with \( i = 2 \). Return to Step 4.

Comment 5. When the loop in Steps 4–5 is completed, we have determined a grid sequence \( h_1, h_2, h_3, \ldots, h_p \), so that \((10)\) is satisfied for each \( i = 1, 2, \ldots, p - 1 \), so that \( h_i/h_{i+1} \) is a power of 2 for each \( i = 3, 4, \ldots, p - 1 \), and so that the “jump” from grid \( h_i \) to \( h_{i+1} \) is as large as possible, \( i = p - 1, p - 2, \ldots, 4 \). Incidentally, because of the cubic convergence of RQI, often in practice \( p = 3 \). Note that \( i = 2 \) the first time through Step 6.

Step 6. Let \( \lambda^{(i)} \) denote the Rayleigh quotient of the final approximation on grid \( h_i \) and let \( x^{(0)} \) denote the result of applying \( I^{h_i} \) to this approximate eigenvector. Perform one outer loop inverse iteration according to

\[
(x^{(1)}, A - \lambda^{(i)} B)x^{(1)} = h_{i+1}Bx^{(0)}.
\]

(This is not strictly RQI, although we refer to it as such for our convenience.)

Comment 6. The computation of \( x^{(1)} \) may be done either directly by a factorization scheme or indirectly by some iteration method; see Sections 4 and 6. Note that the mesh refinement process can thus be viewed as a means for linearizing the problem safely in the sense of providing an initial guess that requires only one linearization per grid.

Step 7. Replace \( x^{(1)} \) by \( x^{(1)}/\|x^{(1)}\|_{h_{i+1}} \). If \( i < p \), increment \( i \) by one and go to Step 6. Otherwise, output \( R(x^{(1)}) \) and \( x^{(1)} \) and stop.

Comment 7. Several tests may be required here to insure that the results are appropriate. For example, examination of the problem behavior may be observed on grid \( 2h_p \) to insure that the truncation error estimates are acceptable and that \( x^{(1)} \) on grid \( h_p \) is sufficiently accurate. This is facilitated by using multigrid as the inner loop iteration but is expensive when other techniques are used to solve \((11)\).

The assumption in \((3)\), that \( \lambda_1 \) is simple, is not strictly necessary, although for multiple \( \lambda_1 \) we can expect that the discretization in effect splits \( \lambda_1 \) into two or more very close eigenvalues. Thus, the mesh refinement process will be confused in its determination of \( c_1 \) and \( c_2 \), unless a precaution is taken to recognize the condition that \( |\lambda_1 - \lambda_2| \) is tending to zero. An alternative is to implement simultaneous iteration by carrying several vectors at once in order to approximate several of the lowest eigenvalues in \((3)\). The Ritz steps should then be performed after each application of \((11)\), although the solution process in connection with \((11)\) must now account for several vectors together. In terms of the multigrid iteration, this will result in a conservative algorithm by requiring that the smoothing scheme account for all of the vectors at once. In terms of the mesh refinement process, \((7)\) must then be modified to account for the subspace truncation error estimates which must then be incorporated into the algorithm.

The mesh refinement strategy developed here is very efficient, but perhaps a greater attribute is the control that it provides in the solution process. Thus, as long as the target solution is even vaguely apparent on the coarsest grid, this strategy allows for a complete examination of the coarse grid problem to direct the procedure on the finer grid to the proper target. It is probably best then to think of
this approach as a means of providing an acceptable starting guess for the fine grid iteration, a perennial problem for fast converging processes such as RQI.

This mesh refinement strategy is independent of the way in which (11) is to be solved. Certainly for two-point boundary value problems, direct factorization is usually best. A numerical example in this connection is given in Section 6. But for more complex problems (e.g., $d > 2$), it is often more efficient to use a multigrid approach. It has been suggested [6] that multigrid methods for solving inverse iteration equations (first proposed in [13]) will be unacceptably slow because of the inherent numerical conditioning of the defining equations. However, multigrid methods are apparently insensitive to the condition of the linear system and in fact work remarkably well in application to (11), as we shall see in Section 6.

4. The Multigrid Process. There have been several recent and essentially unsuccessful attempts to develop algorithms for solving (1) based upon defining outer-loop RQI and using an innerloop iterative scheme like conjugate gradients. Although such methods may need suitable modification to account for possible indefiniteness of the RQI defining matrix, the major difficulty with these methods is rather the matrix condition. These methods fail uniformly for most eigenproblems, especially those stemming from differential operators. The potential for success with the multigrid process (cf. [5]) therefore lies in its insensitivity to the condition of the linear problem and its ability to solve most problems in $O(n(h))$ arithmetic operations.

The correction scheme [5] is a slightly more efficient technique than FAS and other multigrid approaches since it makes use of the linearity of the differential operator. Moreover, it is computationally advantageous for the multigrid process in general if the operator is also positive definite or, more precisely, if the operator approximation on the coarsest grid used in the grid cycling scheme is nonnegative definite. This insures that the relaxation iterations do not inordinately magnify the error components in the eigenvectors belonging to negative eigenvalues and thus eliminates the need to change to a different and, perhaps, direct solution process on the coarsest grid. These two conditions, namely, linearity and coarse grid nonnegative definiteness, can be guaranteed by proper choice of the shifts in (11).

Another important point is that the solution of (11) on grid $h_{i+1}$ should proceed by first starting the multigrid iterations on grid $h_i/2$ using the approximation interpolated from grid $h_i$. The multigrid scheme would then progress in turn on grids $2^{-2}h_i, 2^{-3}h_i$, and so forth, until grid $h_{i+1}$ is reached. Unless the solution is acceptable at this point, the cycling would then begin with the use of grids $h_i, 2^{-1}h_i, 2^{-2}h_i, \ldots, h_{i+1}$. This suggests that the residual equation be solved by direct means on grid $h_i$, provided it is not too fine. However, care should be taken here since the defining matrix on grid $h_i$ is now numerically singular and will result in a correction that will dominate the approximation on grid $h_{i+1}$. One of several ways to avoid this behavior is to replace the shift by the Rayleigh quotient of the fine grid approximation, assuming that the relaxation sweeps have sufficiently damped out the higher eigenvalue contributions. In the experiments reported in Section 6, we succeeded by simply ignoring this problem. We were able to do this because the components that were magnified, because of the lack of positive definiteness, were very close to the eigenvectors being sought. For more complex problems, it is
suggested that the Rayleigh quotient be computed after a few relaxation sweeps on
grid $h_{i+1}$, that it be used as the new RQI shift, and that the grid $h_i$ correction be
computed by a technique that does not require positive definiteness (e.g., by
factorization).

5. Theory for Variational Methods. The purpose of this section is to develop a
theoretical footing for the mesh refinement process developed in Section 3. Since
the method relies on estimates of upper bounds for $|h_1 \lambda_1 - h_{i+1} \lambda_1|$, then, unless
rigorous bounds are known from analysis, a complete theory would require
assumptions on the existence of positive lower bounds for these quantities. Indeed,
without proper safeguards, accidentally small values of these quantities on coarse
grids could easily trip up the refinement process. This is one reason for possibly
implementing a conservative fudge factor in (10) or using more than two initial
grids to improve the estimate for $c_1$. Nevertheless, we will assume $c_1$ has been
properly estimated and focus our attention on motivating the mesh transition
expression depicted in (10). For simplicity, we assume a variational formulation of
the discretization in (4a) with a setting somewhat more general than that described
in Section 2.

Suppose $\hat{\mathcal{A}} : D(\hat{\mathcal{A}}) \rightarrow H$ and $\mathcal{B} : D(\mathcal{B}) \rightarrow H$ are selfadjoint linear operators with
domains $D(\hat{\mathcal{A}}) \subset D(\mathcal{B})$ that are dense subsets of the real separable Hilbert space $H$. Suppose also that $\mathcal{B}$ is positive definite on $D(\mathcal{B})$. Assume that $S_1 \subset S_2 \subset \cdots \subset D(\hat{\mathcal{A}})$ represents an infinite sequence of closed subspaces for which
\[ \lim_{n \to \infty} S_n = \bigcup_{n=1}^{\infty} S_n \] is dense in $H$. We assume, for definiteness, that $\dim S_n = n$. Let
$\mathcal{P}_n : H \rightarrow S_n$ denote the canonical orthogonal projector of $H$ onto $S_n$ and
write $A_n = \mathcal{P}_n \mathcal{A} \mathcal{P}_n$ and $B_n = \mathcal{P}_n \mathcal{B} \mathcal{P}_n$. Then the eigenvalues of
\[ A_n x = \lambda B_n x \]
are real and may be written, counting multiplicities, as
\[ \lambda_{1n} < \lambda_{2n} < \cdots < \lambda_{nn}. \]
The corresponding $B$-unit eigenvectors are written as
\[ u_{1n}, u_{2n}, \ldots, u_{nn}. \]
(By $B$-unit vectors we mean vectors of unit length in the inner-product $\langle x, y \rangle_B = \langle \mathcal{B}x, y \rangle$, $x, y \in D(\mathcal{B})$. The corresponding $\mathcal{B}$-norm is given by $\|x\|_B = \langle \mathcal{B}x, x \rangle^{1/2}$.)

Suppose $\hat{\mathcal{A}} x = \lambda \mathcal{B} x$ has a complete set of $\mathcal{B}$-orthonormal eigenvectors
\{ $u_1, u_2, \ldots$ \} that correspond to the eigenvalues $\lambda_1 < \lambda_2 < \ldots$ (assuming, for
simplicity, that the spectrum is bounded from below). Define
\[ \epsilon_n = \lambda_{1n} - \lambda_1 \]
for $1 < i < n$ and note that our variational setting implies that $\epsilon_n$ is a nonincreasing
sequence of reals bounded from below by zero. We, naturally, assume that
\[ \lim_{n \to \infty} \epsilon_n = 0 \] and, further, that $\lambda_1$ is simple. We also require
\[ \lim_{n \to \infty} \lambda_{2n} = \lambda_2. \] Let
\[ m = \lambda_2 - \lambda_1, \quad m_n = \lambda_{2n} - \lambda_{1n}, \] and $R(x) = \langle \hat{\mathcal{A}} x, x \rangle / \langle \mathcal{B} x, x \rangle$, $x \in D(\mathcal{B})$. In what follows, note that $R(x) = \langle \hat{\mathcal{A}} x, x \rangle / \langle \mathcal{B} x, x \rangle$, whenever $x \in S_n$.

The first lemma provides an estimate for the error in $x \in S_n$ as an approxima-
tion to $u_{1n}$ in terms of the error $R(x) - \lambda_{1n}$. 
**Lemma 1.** Let \( n > 1 \) and \( x \in S_n \) such that \( \varepsilon = R(x) - \lambda_{1n} < m_n/2 \). Then

\[
\langle x, u_{1n} \rangle_{B_n} \neq 0 \quad \text{and} \quad
\left\| \frac{x}{\langle x, u_{1n} \rangle_{B_n}} - u_{1n} \right\|_{B_n}^2 \leq \frac{\varepsilon}{m_n - \varepsilon}.
\]

**Proof.** Write \( x = \langle x, u_{1n} \rangle_{B_n} (u + \delta h) \), where \( h \in S_n \), \( \langle B_n u_{1n}, h \rangle = 0 \), and \( \langle B_n h, h \rangle = 1 \). Then \( \delta^2 \) is the left side of the inequality in (15). This inequality now follows from the observation

\[
\varepsilon = R(x) - \lambda_{1n} = \frac{\lambda_{1n} + \delta^2 \langle A_n h, h \rangle}{1 + \delta^2} - \lambda_{1n} = \frac{\delta^2}{1 + \delta^2} \langle (A_n - \lambda_{1n} B_n) h, h \rangle > \frac{m_n \delta^2}{1 + \delta^2}. \quad \square
\]

The next lemma yields upper bounds for the error after one iteration of (11) performed on \( S_n \).

**Lemma 2.** Let \( \lambda \) satisfy \( 0 < \lambda - \lambda_{1n} < m_n/2 \). Given \( x \in S_n \) such that \( \langle x, u_{1n} \rangle_{B_n} \neq 0 \), let \( y \in S_n \) be the solution to

\[
(A_n - \lambda B_n) y = B_n x.
\]

We then have

\[
R(y) - \lambda_{1n} \leq \frac{\delta^2 (\lambda - \lambda_{1n})^2 m_n}{(\lambda_{2n} - \lambda)^2 + \delta^2 (\lambda - \lambda_{1n})^2},
\]

where \( \delta = \|x/\langle x, u_{1n} \rangle_{B_n} - u_{1n}\|_{B_n} \).

**Proof.** We first assume \( B = I \), the identity on \( H \), so that \( B_n = \Phi_n \) and \( B_n/S_n = \Phi / S_n \). Assuming, without loss of generality, that \( \lambda = 0 \) so that \( \lambda_{1n} < 0 < \lambda_{2n} \), then

\[
R(y) - \lambda_{1n} = \frac{\langle A_n y, y \rangle}{\langle y, y \rangle} - \lambda_{1n} = \frac{\langle A_n A_n^+ x, A_n^+ x \rangle}{\langle A_n^+ x, A_n^+ x \rangle} - \lambda_{1n},
\]

where \( A_n^+ \) is the inverse of \( A_n / S_n \). Hence, writing \( x = \langle x, u_{1n} \rangle_{B_n} (u_{1n} + \delta h) \), as in the proof of Lemma 1, we have

\[
R(y) - \lambda_{1n} = \frac{\delta^2 \lambda_{1n}^2 \langle A_n^+ h, h \rangle - \lambda_{1n} \delta^2 \langle A_n^+ h, A_n^+ h \rangle}{\langle h, h \rangle + \delta^2 \lambda_{1n}^2 \langle A_n^+ h, A_n^+ h \rangle}.
\]

The maximum of this quantity over unit \( h \in \{ u_{1n} \}^\perp \cap S_n \) occurs as a zero of the gradient of the right-hand side of (18), considered as a function of this \( h \). But this implies that \( h \) is the zero of a second-degree polynomial in \( A^+ \). Hence, \( h \) must be a linear combination of at most two eigenvectors of \( A^+ \) and, therefore, \( A \). Writing \( h = \alpha_i u_{in} + \alpha_j u_{jn} \), where \( \alpha_i^2 + \alpha_j^2 = 1 \), \( i, j > 2 \), and setting \( \alpha = \alpha_2^2 \), we have

\[
R(y) - \lambda_{1n} \leq \frac{\delta^2 \lambda_{1n}^2 \left[ (\lambda_{in} - 1 - \lambda_{jn} - 1) \alpha + \lambda_{in} - 1 - \lambda_{jn} - 1 \right]}{1 + \delta^2 \lambda_{1n}^2 \left[ (\lambda_{in} - 2 - \lambda_{jn} - 2) \alpha + \lambda_{in} - 2 \right]}\]

for some \( i, j > 2 \) and \( 0 < \alpha < 1 \). We must now maximize this expression over \( i, j \), and \( \alpha \), which we first do for \( \alpha \). Noting that the right-hand side is simply a quotient
of two linear terms in $\alpha$, the maximum over $\alpha$ must occur either at $\alpha = 0$ or at $\alpha = 1$. Thus,

$$R(y) - \lambda_{1n} \leq \max_{2 < i < n} \frac{\delta^2 \lambda_{1n}^2 (\lambda_{in} - \lambda_{1n})}{\lambda_{in}^2 + \delta^2 \lambda_{1n}^2}. \quad (19)$$

If we view $\lambda_i$ as an arbitrary real in the interval $[\lambda_{2n}, \lambda_{mn}]$, then the maximum in (19) is attained at either $\lambda_{mn}$, $\lambda_{2n}$, or $\lambda_{1n} \pm \sqrt{\lambda_{1n}^2 - \delta^2}$. The latter are the zeros of the derivative of the right-hand side, considered as a function of $\lambda_i$. But $\lambda_{1n} \pm \sqrt{\lambda_{1n}^2 - \delta^2} < \lambda_2$, so the maximum in (19) is attained at $i = 2$ and (17) now follows.

The case for general $\mathcal{B}$ rests on establishing (17) by performing a Choleski factorization of $B_n/S_n$ and transforming (12) into a symmetric eigenproblem with $B_n/S_n$ becoming the identity. This is straightforward and is therefore omitted. □

**Theorem 1.** Let $N$ be given so that $\varepsilon_k < m_i/2$ for all $l, k \geq N$. Suppose, for a given $k \geq N$, that an $x \in S_k$ has been determined so that

$$|\lambda_{1k} - R(x)| < \varepsilon_k.$$ 

Let $l > k$ satisfy the transition condition

$$\frac{(2\varepsilon_k)^3 m_i}{(m_i - 2\varepsilon_k)^3 + (2\varepsilon_k)^3} \leq \varepsilon_l. \quad (20)$$

Then $y$, determined by (16) with $n = k$ and $\lambda = R(x)$, satisfies

$$|\lambda_{1l} - R(y)| < \varepsilon_l.$$

That is, one iteration is sufficient for convergence to $\lambda_{1l}$ to within the truncation error $\varepsilon_l$.

**Proof.** With $\lambda = R(x)$, note that

$$0 < \lambda - \lambda_{1l} \leq \lambda - \lambda_1 < \lambda - \lambda_{1k} + \lambda_{1k} - \lambda_1 < 2\varepsilon_k.$$

Hence, by Lemma 1, viewing $x$ as an element of $\delta_l$ yields

$$\left\| \frac{x}{\langle x, u_{1l} \rangle_{B_l}} - x_{1l} \right\|_{B_l}^2 < \frac{2\varepsilon_k}{m_i - 2\varepsilon_k}.$$ 

Hence, by Lemma 2,

$$R(y) - \lambda_l \leq \frac{(2\varepsilon_k/m_i - 2\varepsilon_k)(2\varepsilon_k)^2 m_i}{(m_i - 2\varepsilon_k)^2 + (2\varepsilon_k/m_i - 2\varepsilon_k)(2\varepsilon_k)^2} < \frac{(2\varepsilon_k)^3 m_i}{(m_i - 2\varepsilon_k)^3 + (2\varepsilon_k)^3}. \quad (20)$$

The theorem follows from this and (20). □

**Remark 1.** For sufficiently large $N$, the expression in (20) is essentially the same as

$$\frac{(2\varepsilon_k)^3}{m_i^2} < \varepsilon_l.$$
With $e_k = c_k h_k^m$, this is just the mesh refinement transition condition depicted in (10).

Remark 2. The theory of this section can be applied to $\lambda_i$, $i > 1$, with a few obvious modifications.

Remark 3. There are several special contexts to which the general assumptions of this section apply, such as eigenproblems for integral operators (cf. [3]) and differential operators (cf. [6]).

6. Numerical Results. In this section, we report on the results of experiments with the methods presented here applied to four specific examples: two two-point boundary value eigenproblems ($d = 1$) and two partial differential eigenproblems ($d = 2$). Note that $B = I$ in each case. Since the performance of these methods is invariant under similarity transformations, it is sufficient, from a numerical point of view, to test these problems with $B = I$. That is, a Choleski decomposition of $hB = hLh^T$ can be used to transform the case in which $hB$ is not the identity to one in which it is. The numerical performance of the methods of this paper is invariant under such a transformation. Note that this viewpoint is in the same spirit as that in using only diagonal matrices (cf. [12]) to test the Lanczos process. The first two examples are the following two-point boundary value eigenproblems:

E1. (Simple Harmonic Motion Equation).

\[- x'' = \lambda x,\]
\[x(0) = x(1) = 0,\]
\[\int_0^1 x^2(t) \, dt = 1,\]

(actual $\lambda_1 = \pi^2 = 9.8969044$).

E2. (Mathieu's Equation).

\[- x'' + 20\pi^2 \cos(2\pi t) x = \lambda \pi^2 x,\]
\[x(0) = x(1) = 0,\]
\[\int_0^1 x^2(t) \, dt = 1,\]

(actual $\lambda_1 = -13.9365525$).

To, in effect, ignore the special properties of these problems, we have implemented them in a rather naive way. For example, we discretized both by standard central differences on a uniform grid on $[0, 1]$ (so that $m = 2$ in (7)), we made no use of any accelerating techniques such as extrapolation, and we restricted $h_1$, $h_2$ to simple linear interpolation. We also focused on the computation of $\lambda_1$, although higher eigenvalues may be computed by these methods. All tests were done on a CYBER 172.

The mesh refinement process was first implemented by solving for each RQI iterate on each grid directly by elimination. Table 1 depicts $h_1 \hat{\lambda}_1$ and $h_2 \hat{\lambda}_1$ (which are the computed values of $h_1 \lambda_1$ and $h_2 \lambda_1$, respectively, with $h_1 = 2^{-4}$ and $h_2 = 2^{-5}$), the estimate $\hat{c}_1 = 16^2 |h_1 \hat{\lambda}_1 - h_2 \hat{\lambda}_1|$, the predicted acceptable mesh size $h_3$ for which $|h_3 \lambda_1 - \lambda_1| < \text{tol}/2$ (here we use tol = $2^{-18}$), the approximation $h_3 \hat{\lambda}_1$ computed via one RQI on grid $h_3$, and $r = |h_3 \hat{\lambda}_1 - \lambda_1| \text{tol}^{-1}$. The erroneous digits in the
eigenvalue approximations are underlined. Note that success is indicated by the condition that \( r < 1 \).

This choice of tol(\( \approx 3.8 \times 10^{-6} \)) was made for convenience in our tests. Note that the relative error in each \( \lambda_1 \) is therefore less than \( 5 \times 10^{-7} \).

Convergence on the fine grid was achieved, as expected, for each problem by the first RQI iterate. In our experiments, we were actually able to do much better than is indicated. In fact, we were able to jump from grid \( h_1 = 2^{-2} \) to \( h_2 = 2^{-14} \) for both problems without incurring the need for additional RQI steps. This is not surprising since the analysis of RQI is naturally pessimistic, but it can be fortunate since it will tend to counteract the errors in determining an acceptable value for \( c_1 \).

| Example | \( h_1 \hat{\lambda}_1 \) | \( h_2 \hat{\lambda}_1 \) | \( c_1 \) | \( h_3 \) | \( h_3 \hat{\lambda}_1 \) | \( r = |h_3 \hat{\lambda}_1 - \lambda_1| \text{tol}^{-1} \) |
|---------|----------------|----------------|--------|-------|----------------|----------------|
| E1     | 9.838          | 9.8617         | 6      | 2^{-11}| 9.8696026      | .47            |
| E2     | -14.022        | -13.9575       | 16     | 2^{-12}| -13.9365538    | .33            |

| Example | No. of Cycles | \( h_3 \hat{\lambda}_1 \) | \( r = |h_3 \hat{\lambda}_1 - \lambda_1| \text{tol}^{-1} \) |
|---------|---------------|----------------|----------------|
| E1     | 0             | 9.8696027      | .45            |
| E2     | 0             | -13.9365482    | 1.13           |
|        | 1             | -13.9365496    | .76            |

The next set of experiments involved the same technique as before, except that the problem on grid \( h_3 \) was solved by a fixed correction scheme multigrid method [5]. Thus, the only difference between these and the first set of tests was that the direct process of solving for the RQI iterate on grid \( h_3 \) was replaced by multigrid cycles initiated on grid \( h_2/2 \). (The multigrid scheme was of fixed type in that the cycling was predetermined to move progressively from grid \( h_2 \) to \( h_3 \) and back. In progressing to grid \( h_3 \), three Gauss-Seidel relaxation sweeps were used while one was used in the return to grid \( h_2 \). Although this is a fairly primitive multigrid scheme, it serves to illustrate the performance of methods of this type.) Each time the multigrid iteration arrived at grid \( h_3 \), the Rayleigh quotient of the iterate was computed and compared to the true \( \lambda_1 \) for test purposes to determine performance. (Of course, a computable convergence criteria is necessary in practice. A safe test is to compute the norm of the gradient of \( R(x) \) which can be used to bound the error in \( x \).)

Table 2 depicts the results of tests on both equations. The cycle count was initially set to zero when the multigrid iteration first arrived at grid \( h_3 \) and was
incremented by one each time it returned from coarse grid cycling. Note the remarkable success of this scheme especially for equation E1. In fact, this suggests that combined mesh refinement multigrid iteration will solve most eigenvalue problems to which it applies in an equivalent of just a few fine grid matrix multiplies, regardless of whether or not \( \mathcal{B} = 9 \). This is underscored by the fact that a relatively major expense in producing an acceptable eigenvalue by this approach is in the computation of the Rayleigh quotient of the fine grid approximation.

The last two examples are two-dimensional partial differential boundary value eigenproblems:

E3. *(Helmholtz Equation in the Unit Circle).*

\[-\Delta x = \lambda x \quad \text{in } \Omega = \{(s, t): s^2 + t^2 < 1\},\]

\[x = 0 \quad \text{on } \partial \Omega = \{(s, t): s^2 + t^2 = 1\},\]

\[\int_{\Omega} \int x(s, t) \, ds \, dt = 1,\]

(actual \( \lambda_1 \approx 5.78306 \)).

E4. *(Helmholtz Equation in an Egg-Shaped Region).*

\[-\Delta x = \lambda x \quad \text{in } \Omega = \{(s, t): s^2 + t^2 < 1, \ t < 0\} \cup \{(s, t): 4s^2 + t^2 < 4, \ t > 0\},\]

\[x = 0 \quad \text{on } \partial \Omega = \{(s, t): s^2 + t^2 = 1, \ t < 0\} \cup \{(s, t): 4s^2 + t^2 = 4, \ t > 0\},\]

\[\int_{\Omega} \int x(s, t) \, ds \, dt = 1,\]

(actual \( \lambda_1 \approx 4.2080 \)).

The tests with these examples were with multigrid only. The tol values were chosen to reflect different criteria of accuracy so that the eigenproblem for E4 is larger and produces greater accuracy. In each case, \( h_1 = 2^{-1} \) so that \( h_2 = 2^{-2} \). Note that the performance, as expected, is similar to that of the method applied to the one-dimensional examples E1 and E2.

These examples are typical of the behavior we have observed in experiments with the mesh refinement/multigrid process. In all of the tests we have run, the eigenvalue approximation was acceptable after one multigrid cycle from the finest grid. In most cases, it was acceptable even before the cycle.

**Table 3. Mesh Refinement / Multigrid**

*(with coarse grid elimination)*

| Example | \( h_1 \hat{\lambda}_1 \) | \( h_2 \hat{\lambda}_1 \) | \( \hat{c}_1 \) | tol | \( h_3 \) | No. of Cycles | \( h_1 \hat{\lambda}_1 \) | \( r = |h_1 \hat{\lambda}_1 - \lambda_1|^{10^{-1}} \) |
|---------|----------------|----------------|-------|-----|------|------------|----------------|------------------|
| E3      | 5.439          | 5.681          | 1.3   | 10^{-2} | 2^{-4} | 0          | -5.778         | .5               |
| E4      | 4.0162         | 4.1524         | .73   | 10^{-3} | 2^{-5} | 0          | -4.2071        | .9               |

**Acknowledgements.** The author wishes to thank Achi Brandt, Paddy Prenter, and John Locker for several helpful discussions, Steve Schaffer and John Ruge for the
numerical implementation and tests reported in the last section, and the referee for suggesting several improvements in the exposition.

Department of Mathematics
Colorado State University
Fort Collins, Colorado 80523


