JOURNAL OF COMPUTATIONAL PHYSICS 44, 60–83 (1981)

Calculating Eigenvalues and Eigenfunctions Using an Interior Constraint*

JAMES L. BLUE† AND CHARLES L. WILSON‡

†*Scientific Computing Division, and*
‡*Electron Devices Division, National Bureau of Standards, Washington, D.C. 20234*

Received January 5, 1981

A new method for calculating eigenvalues and eigenfunctions of elliptic operators is presented. An interior constraint is used to allow reliable convergence to any desired eigenfunction. The method has been implemented in a portable Fortran computer program which features adaptively generated triangulations in two dimensions, and which uses multi-level iteration. The program has been used to calculate efficiently eigenvalues and eigenfunctions on singly- and multiply-connected regions, with internal and boundary singularities.

1. INTRODUCTION

We present a method for numerical calculation of eigenfunctions and eigenvalues of self-adjoint elliptic operators. The method has been implemented in a portable Fortran computer program. Important features of the method include reliable convergence to any eigenvalue and its eigenfunction. No initial guess at the eigenfunction is needed.

The method has been implemented with a general-purpose two-dimensional elliptic solver which uses linear finite elements on triangles. Any general elliptic solver could be used, with only minor modifications. Our package provides interactive graphical display of eigenfunctions and triangulations.

Our package generates a sequence of approximations to an eigenfunction. Each approximation is a finite element solution on a triangulation. Errors in each approximate eigenfunction are estimated and used to generate a refined triangulation for the next approximation.

Each eigenvalue and its associated eigenfunction are found using a formulation which requires minimizing a one-parameter function. We have found this to be more reliable than the widely recommended method of inverse iteration.

0021-9991/81/110070–14$02.00/0

2. Formulation

We wish to solve the two-dimensional self-adjoint eigenvalue problem

$$\begin{aligned} -\nabla \cdot (a\nabla u) + qu &= \lambda pu \qquad \text{in } \Omega, \\ u &= 0 \qquad \text{on } \partial\Omega_1, \\ \frac{\partial u}{\partial n} &= 0 \qquad \text{on } \partial\Omega_2 = \partial\Omega - \partial\Omega_1 \end{aligned} \tag{1}$$

for a few eigenvalue–eigenfunction pairs λ, u. The coefficients a, q, and p can be functions of x and y. We assume $a > 0$ and $p > 0$ in Ω. Boundary conditions are piecewise homogeneous Dirichlet and Neumann; the Neumann portion is optional. The elliptic solver we use restricts the domain Ω to be a two-dimensional bounded region, with boundary $\partial\Omega$ consisting of straight line segments and circular arc segments.

Problem (1) has a countable set of real eigenvalues λ_i, $\lambda_1 \leqslant \lambda_2 \leqslant \lambda_3 \leqslant \cdots$, each with an associated eigenfunction u_i. The u_i may be multiplied by any scale factors; we assume the normalization $\iint u_i^2 p \, dx \, dy = 1$. If $\lambda_i \neq \lambda_j$, u_i and u_j are orthogonal [1]:

$$\iint u_i u_j p \, dx \, dy = 0, \qquad \lambda_i \neq \lambda_j.$$

If $\lambda_i = \lambda_j$, new linear combinations of u_i and u_j can be redefined that are orthogonal. We assume this, so that finally

$$\begin{aligned} \iint u_i u_j p \, dx \, dy &= 0, \qquad i \neq j, \\ &= 1, \qquad i = j. \end{aligned}$$

Generally only one or a few of the first eigenvalue–eigenfunction pairs are desired.

Let S be the space of functions which are continuous functions of x and y, have piecewise continuous first derivatives, and obey the Dirichlet boundary condition on $\partial\Omega_1$. The Rayleigh quotient $R(v)$ is defined for any v in S.

$$R(v) = \frac{\iint [a|\nabla v|^2 + qv^2] \, dx \, dy}{\iint pv^2 \, dx \, dy}.$$

The eigenfunctions of (1) are complete for functions in S, so we may expand v as

$$v(x, y) = \sum_{k=1}^{\infty} a_k u_k(x, y).$$

After integrating by parts, using the boundary conditions and the orthonormality of the eigenfunctions, we obtain

$$R(v) = \sum_{k=1}^{\infty} \alpha_k^2 \lambda_k \Big/ \sum_{k=1}^{\infty} \alpha_k^2.$$

Therefore $\lambda_1 = \min_{v \in S} R(v)$, and $\lambda_k = R(u_k)$ for all k.

One way to approximate λ_1 and u_1 is to choose a set of basis functions ϕ_k, $k = 1, 2, \ldots, N$, and let

$$v_1(x, y) \approx \sum_{k=1}^{N} \xi_k \phi_k(x, y).$$

An "optimal" set of coefficients $\{\xi_k\}$ may be found by minimizing $R(v_1)$ with respect to the N ξ_k's. The coefficients are optimal in the sense that they are the set giving the best approximation to λ_1 for the given set of basis functions. The quality of the approximate eigenvalue and eigenfunction depends upon N and upon the choice of the $\{\phi_k\}$. A poor set of basis functions will necessitate a large N to obtain satisfactory accuracy, especially in the eigenfunction.

Similar results hold for higher eigenvalue–eigenfunction pairs, but the trial function v must obey orthogonality conditions as well as the Dirichlet boundary condition. The minimum condition,

$$\lambda_j = \min_{v \in S} R(v),$$

holds only if $\iint p v_j v_k \, dx \, dy = \alpha_k = 0$ for $k = 1, 2, \ldots, j-1$.

In the remainder of this paper, we let the basis functions $\phi_k(x, y)$ be linear finite elements on triangles. Let Ω be triangulated, with vertices (x_k, y_k), $k = 1, 2, \ldots, N$. Then $\phi_k(x_k, y_k) = 1$, $\phi_k(x_j, y_j) = 0$ if $j \neq k$, and ϕ_k is linear on each triangle. Only basis functions obeying the Dirichlet boundary condition are included. We denote finite element superpositions by a superscript h. Let

$$u^h(x, y) = \sum_{k=1}^{N} \beta_k \phi_k(x, y)$$

and use the standard finite element formulation of (1), as in [2]. This gives a matrix eigenvalue problem of the form

$$Ab = \lambda^h B b,$$

where b is an N-vector $(\beta_1, \beta_2, \ldots, \beta_N)^T$ and A and B are $N \times N$ sparse symmetric matrices, typically with 5 to 9 nonzero elements per row. A is positive definite. This

problem has N real eigenvalues λ_i^h, $\lambda_1^h \leqslant \lambda_2^h \leqslant \cdots$, each with an associated eigenvector c_i. The orthonormality of the c_i is

$$c_i^T B c_j = \begin{cases} 0, & i \neq j, \\ 1, & i = j. \end{cases}$$

The lower of the λ_i^h will approximate eigenvalues of (1), but the higher ones will not. Since u^h is in S,

$$\lambda_1 \leqslant \lambda_1^h.$$

The Rayleigh quotient for the matrix problem is

$$R^h(b) = \frac{b^T A b}{b^T B b}.$$

An argument similar to the one given for the continuous problem shows that

$$\lambda_1^h = \min_b R^h(u^h) = R^h(b_1)$$

and

$$\lambda_k^h = R^h(b_k) \qquad \text{for all } k.$$

The above argument is valid for other elliptic solvers, although the structure of the matrices A and B may be different.

3. Adaptive Triangulation

The error[1] $|\lambda_k - \lambda_k^h|$ depends on N and on how well the triangulation is suited to the eigenfunction u_k. In general, an appropriate triangulation is not known before the form of the eigenfunction is known. Our programs generate an appropriate triangulation adaptively, in the course of calculating an approximate eigenvalue and eigenfunction.

We start with a coarse triangulation, denoted Δ_1, of Ω. This initial triangulation must be fine enough to resolve the desired eigenfunction, but need not be fine enough to provide an accurate representation of the eigenfunction. The adaptive algorithm is as follows:

For $m = 1$ to max

(1) On Δ_m, calculate finite element matrices A_m and B_m, each of order N_m.

(2) Solve $A_m b = \lambda B_m b$ for the desired eigenvalue λ^m and eigenvector b^m. If

[1] Except for $k = 1$, it is possible that u_k^h corresponds to u_j and $j \neq k$. We ignore this minor complication.

$m > 1$, λ^{m-1} and b^{m-1} will be good initial approximations. Two iterative methods will be discussed for finding λ^m and b^m, inverse iteration and a fixed-point method. One step of each iterative method requires generating and solving a system of element equations. We solve these by a multi-level iteration, to be discussed later. The program to do this has evolved from a program of Bank and Sherman [3].

(3) Estimate an error associated with each triangle. The method of estimation is similar to that in [4] and is described in [3].

(4) Produce a finer triangulation, Δ_{m+1}, by refining the triangles with the largest errors. Each is divided into four similar triangles by connecting the midpoints of the sides.

If the triangulations were uniform, with largest triangle side $h_m = h_1 2^{1-m}$, then the maximum errors in λ^m and in u^m are $O(h_m^2)$. For nonuniform triangulations, we expect the errors to be $O(1/N_m)$ if the refinement is reasonable. This behavior can be used to estimate the errors, using two or more successive approximations on triangulations. In practice, one step of Richardson extrapolation of the eigenvalues seems useful for improving the approximate eigenvalue and for estimating the error in the eigenvalue.

4. Multi-Level Iteration

At level m of the adaptive process described in Section 3, when finding the approximate eigenvalue λ^m, a system of linear equations must be solved. Each row of the system corresponds to one vertex; the number of nonzero entries in the row is 1 plus the number of triangles meeting at the vertex. Since the method of refinement does not increase this number, the matrices are increasingly sparse as m increases.

Standard sparse matrix techniques could be used to solve the linear equations, but a multi-level iteration generally is faster [5] and requires less computer storage than direct solution. We can take advantage of having several sets of linear equations, each modeling the same cntinuous problem, on grids which are nested triangulations of the domain.

Iterative methods are common [6, 7] for solving single systems of equations arising from partial differential equations. Their convergence rate may be qualitatively understood by considering a Fourier expansion of the error in an approximate solution. The convergence rate, the factor by which each Fourier component of the error is multiplied at the end of one iteration, is strongly dependent on the component. For the highest components, the rate is typically 0.5; for the lowest components, the rate is $1 - O(h^2)$, where h is the local mesh size [5]. Standard iterative methods therefore damp out the most oscillatory components of the error quickly, and damp out the slowly varying components slowly. Final convergence is governed by the slowly varying components, and is slow.

Multi-level iterations use the linear equations on all of the grids to solve the linear equations on the finest grids. The most oscillatory error is damped quickly by iteration on the finest grid. The next most oscillatory errors are damped by iteration

on the next coarser grid using the linear equations defined on that grid; these Fourier components are the most oscillatory ones on the grid, and so are damped out quickly. The next most oscillatory errors can be damped out quickly on the next coarser grid, and so on. In this way computer time is saved because few of the iterations are done on the finest grid.

A great variety of multi-level iterative schemes is possible; the reader is referred to [3] for details of the scheme used in our calculations.

5. Inverse Iteration

Inverse iteration [8–10] is the standard method for calculating a few eigenvectors and eigenvalues of matrices. Starting with a guess c_0, a sequence c_1, c_2,..., is produced which usually converges to some eigenvector of $Ab = \lambda Bb$.

Given c_i

$$\tilde{c}_i = \frac{c_i}{(c_i^T B c_i)^{1/2}} \qquad \text{(normalization)},$$

$$\mu_i = A^T \tilde{c}_i A \qquad \text{(Rayleigh quotient)}.$$

Solve for c_{i+1}:

$$(A - \mu_i B)\, c_{i+1} = B\tilde{c}_i.$$

If c_i is close enough to some b_j, then the sequence of c's converges cubically to b_j, and the sequence of μ's converges cubically to λ_j.

There are two problems with inverse iteration. The obvious one is that the b_j to which the iteration converges depends on the initial guess c_0. Convergence to a desired eigenvector may not be possible without fairly accurate knowledge of the eigenvector.

The second problem is that when the sequence is converging, the matrices $A - \mu_i B$ become more and more ill-conditioned with increasing i, since the object is to produce a zero eigenvalue of $A - \mu_i B$. If the linear equations for c_{i+1} are solved by Gaussian elimination, the ill-conditioning is not a problem [9]. If the linear equations are solved by a multi-level iteration, as described earlier, convergence can be intolerably slow, or divergence may occur. This is, in part, because 0 is nearly an eigenvalue of $(A_m - \mu_i B_m)$ for the largest m, but 0 is far from being an eigenvalue of $(A_1 - \mu_i B_1)$.

6. The Interior Constraint Method

If $\lambda = \mu$ is a specified constant in (1), then (1) has only the solution $u = 0$, unless λ happens to be an eigenvalue. In order to avoid the trivial solution, we consider a

modified problem. Let (x_F, y_F) be any interior point of Ω, and let Ω' be Ω minus a circle of (small) radius ε centered at (x_F, y_F). On the circle, let $u = 1$. For small ε, the solution will approximate the Green's function, which has a logarithmic singularity at (x_F, y_F). The coefficient of the logarithm will vanish when μ equals an eigenvalue whose associated eigenfunction is nonzero at (x_F, y_F).

If ε is much smaller than the dimensions of the triangulation near (x_F, y_F), the finite element formulation can ignore the integrals over the circle. Thus we let (x_F, y_F) be an interior vertex of the triangulation and require that $\beta_F = 1$. Rather than $Ac = \mu Bc$, we have

$$(A' - \mu B')\, c = f.$$

The matrices A' and B' are the same as A and B except in row F; f is zero except for row F, which is merely the constraint equation $\beta_F = 1$. The above equation is now inhomogeneous, and we may obtain a solution for any value of μ, denoted $c(\mu, F)$. We keep F fixed and use the Rayleigh quotient to define a one-variable function

$$Q(\mu) = R^h(c(\mu, F)).$$

For the lowest eigenvalue,

$$Q(\mu) \geqslant \lambda_1^h.$$

If $\mu = \lambda_j^h$, then the auxiliary boundary condition at F only changes the normalization, so that

$$c(\lambda_j^h, F) = \alpha b_j$$

for some constant, α. Since the Rayleigh quotient is independent of α, $Q(\lambda_j^h) = \lambda_j^h$. Therefore a graph of $Q(\mu)$ versus μ, in the vicinity of λ_1^h, must look like the part of Fig. 1 near $\mu = 5$.

For higher eigenvalues, we also have

$$Q(\lambda_j^h) = \lambda_j^h$$

if $u_j^h(x_F, y_F) \neq 0$, but there may not be a local minimum of $Q(\mu)$ at λ_j^h. We expand $c(\mu, F)$ in eigenvectors about b_j and expand the coefficients in a Maclaurin series. Then to within a normalizing constant we have

$$c(\mu, F) = b_j + (\lambda_j^h - \mu) \sum_{k=1}^{N} \alpha_k(\mu)\, b_k + O(\lambda_j^h - \mu)^2$$

for some set of functions α_k. By inserting this formula into the definition of $Q(\mu)$, using the orthonormality of the eigenvectors, and differentiating, we find $dQ/d\mu = 0$ at $\mu = \lambda_j^h$.

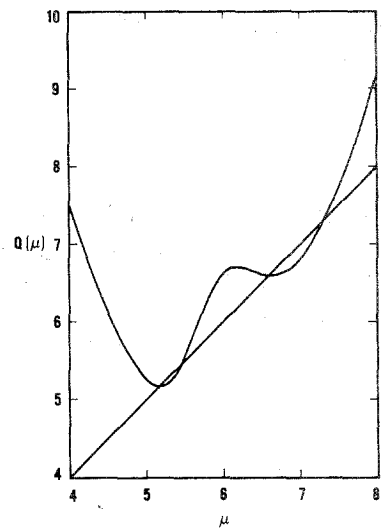

FIG. 1. $Q(\mu)$ vs μ for slit membrane, triangulation of Fig. 2.

At each step of the interior constraint method, the set of linear equations is

$$(A' - \mu B')\, c = f.$$

The eigenvalues of $A' - \mu B'$ are in general not equal to $\lambda_j^h - \mu$. The linear equations do not become singular as $\mu \to \lambda_j^h$, and remain easy to solve with multi-level iteration.

At the mth stage of the adaptive mesh generation procedure of Section 3, it is necessary to find an eigenvalue of $Ac = \lambda Bc$, where we have dropped reference to the level number m.

For $m = 1$, the crudest triangulation, the matrices are small and Gaussian elimination is fast and accurate using programs from the Yale Sparse Matrix Package [11]. A scan of $Q(\mu)$ vs μ may be done cheaply in order to isolate the roots of $Q(\mu) = \mu$ for which $\partial Q/\partial \mu = 0$. After the roots have been isolated, any reliable one-dimensional root finder may be used, and quadratic convergence is easy to obtain.

For $m > 1$, the eigenvalue from the lower level is a good initial guess to give to the root finder.

Newton's method applied to $Q(\mu) - \mu = 0$ gives

$$\mu^{(i+1)} = \mu^{(i)} - \frac{Q(\mu^{(i)}) - \mu^{(i)}}{Q'(\mu^{(i)}) - 1}.$$

Sufficiently near a root, $Q' \approx 0$, Newton's method reduces to resubstitution,

$$\mu^{(i+1)} = Q(\mu^{(i)}),$$

and converges quadratically.

For $m > 1$, the eigenvalue from the lower level is usually a sufficiently good initial guess for resubstitution to converge.

7. An Example

We illustrate our method with an easy elliptic operator on a difficult domain. Another example may be found in [12]. We solve for eigenvalues and eigenfunctions of a vibrating square membrane with side L and with a diagonal slit in it (shown by heavy lines in Fig. 2). The equation is $-\nabla^2 u = \lambda u$, with $u = 0$ on the square and $\partial u/\partial n = 0$ on each side of the slit. Without the slit, the eigenfunctions are proportional to

$$\psi_{nm} = \sin \frac{n\pi x}{L} \sin \frac{m\pi y}{L}$$

with eigenvalues $(n^2 + m^2)\,\pi^2/L^2$, for n and m positive integers. The eigenvalues and eigenfunctions of the slit membrane are not known, except that any ψ_{nm} with $\partial\psi/\partial n = 0$ on the slit, such as ψ_{11}, is an eigenfunction.

We start with the uniform mesh of Fig. 2, which has $N = 28$, plus 16 vertices on the square's boundaries.

$Q(\mu)$ vs μ for $4 \leqslant \mu \leqslant 8$, is shown in Fig. 1, with the mesh of Fig. 2. The first and second λ_i^h are 5.1729 and 6.6085. Successive refinements for the first eigenfunction yield the meshes of Figs. 3a and 3b; those for the second eigenfunction yield Figs.

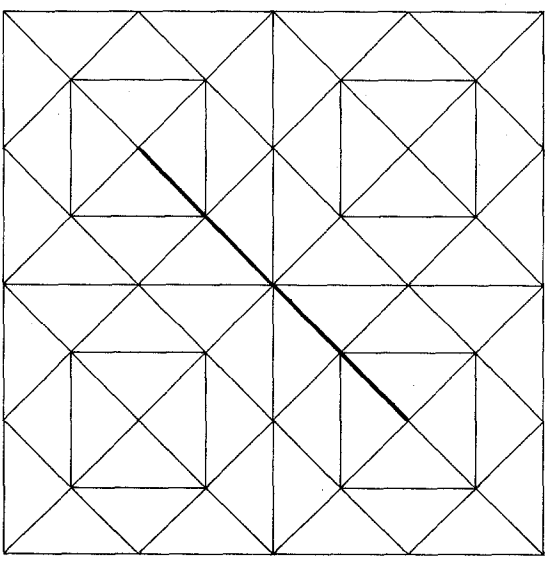

FIG. 2. Initial triangulation for slit membrane.

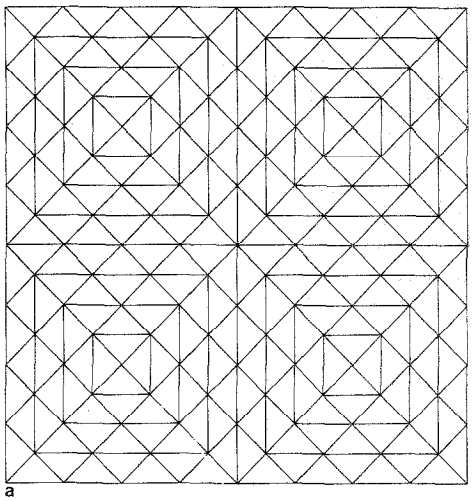

FIG. 3a. First refinement, first eigenfunction.

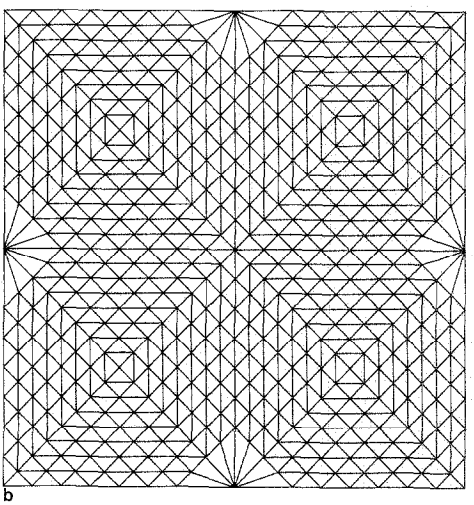

FIG. 3b. Second refinement, first eigenfunction.

TABLE I

m	N	λ_1^h	Error	λ_1^h(ext)	Error
1	28	5.1729	0.2381		
2	120	4.9969	0.0621	4.9433	0.0085
3	484	4.9508	0.0160	4.9356	0.0008

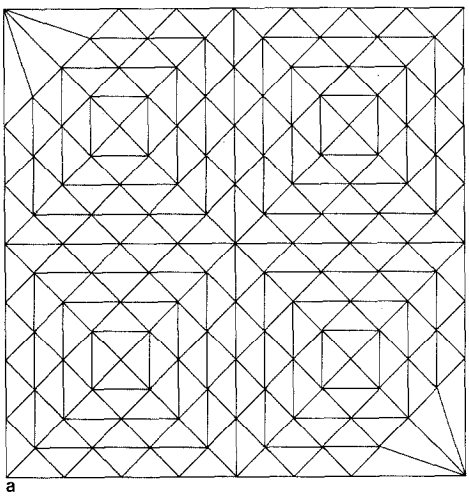

FIG. 4a. First refinement, second eigenfunction.

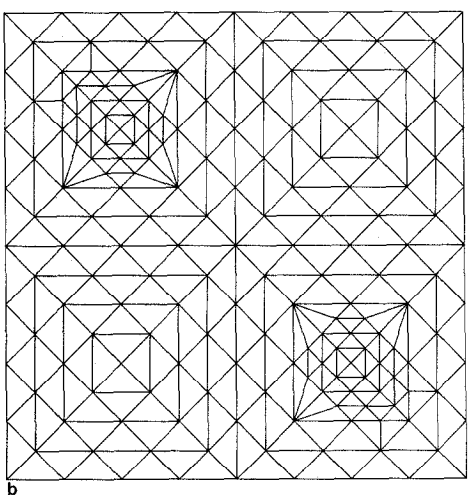

FIG. 4b. Second refinement, second eigenfunction.

TABLE II

m	N	λ_2^h	λ_2^h(ext)
1	28	6.6085	
2	146	6.1283	6.0144
3	192	5.9971	5.5807
4	268	5.9251	5.7432
5	472	5.8611	5.7770

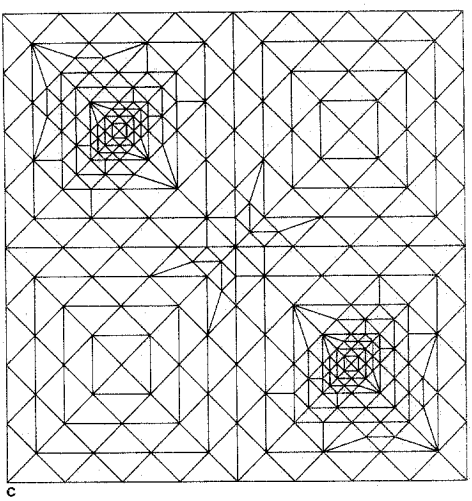

FIG. 4c. Third refinement, second eigenfunction.

4a–d. Figure 5 is a view of the second eigenfunction for the fifth mesh. The results for λ_1^h and λ_2^h are summarized in Tables I and II. The true λ_1 is $\pi^2/2 \approx 4.93480$. The error in λ_1^h behaves approximately as $1/N$. Assuming $1/N$ behavior (justified only for uniform meshes) and extrapolating successive rows, we get λ_1^h (ext) in Table I, with error as shown in the final column. The true λ_2 is not known; λ_2^h (ext) in Table II assumes $1/N$ behavior.

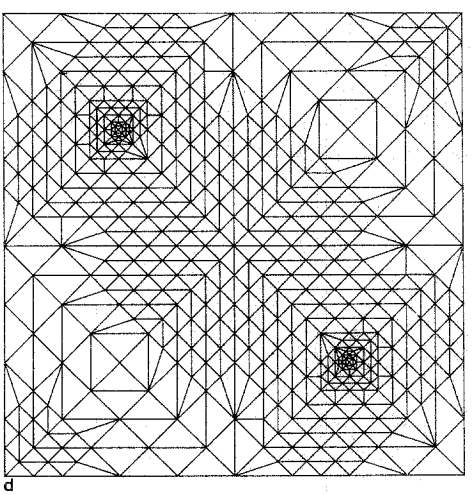

FIG. 4d. Fourth refinement, second eigenfunction.

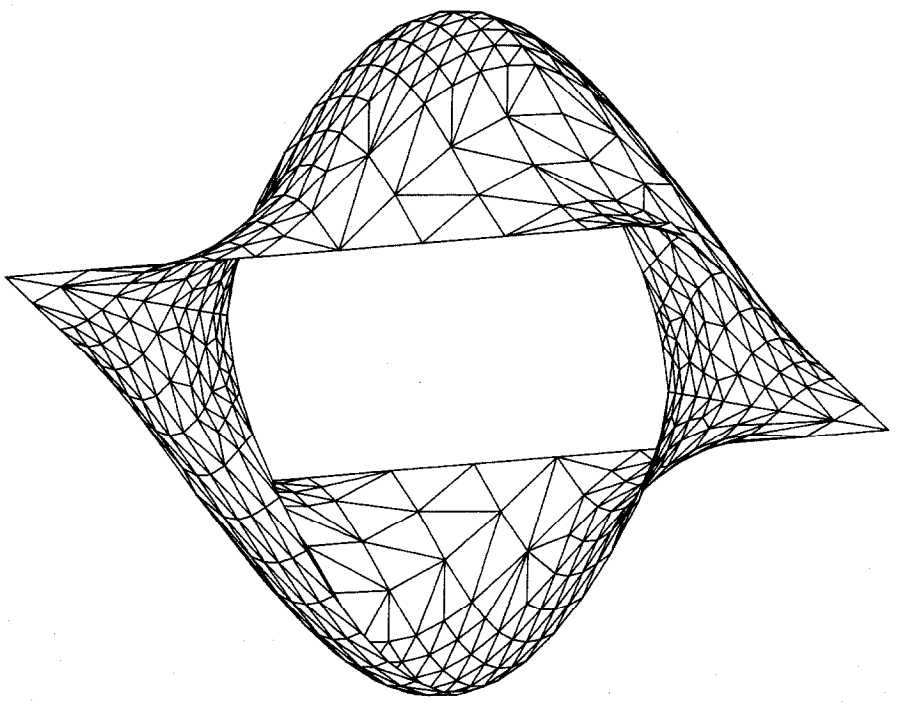

FIG. 5. Second eigenfunction, triangulation of Fig. 4d.

8. CONCLUSION

We have described a method for reliable numerical calculation of eigenvalues and eigenfunctions of elliptic operators. Our implementation, using linear finite elements on triangles, generates triangulations adapted to the eigenfunction being calculated. The program has been used to calculate efficiently eigenvalues and eigenfunctions on singly and multiply connected two-dimensional regions, with internal and boundary singularities.

REFERENCES

1. R. COURANT AND D. HILBERT, "Methods of Mathematical Physics," Interscience, New York, 1953.
2. G. STRANG AND G. J. FIX, "An Analysis of the Finite Element Method," Prentice–Hall, Englewood Cliffs, N. J., 1973.
3. R. E. BANK AND A. H. SHERMAN, "PLTMG Users' Guide," CNA-152, Center for Numerical Analysis, University of Texas at Austin, September 1979.
4. F. BABUSKA AND W. C. RHEINBOLDT, *SIAM J. Numer. Anal.* **15** (1978), 736–754.
5. A. BRANDT, *Math. Comput.* **31** (1977), 333–390.
6. R. S. VARGA, "Matrix Iterative Analysis, "Prentice–Hall, Englewood Cliffs, N. J., 1962.

7. D. YOUNG, "Iterative Solution of Large Linear Systems," Academic Press, New York, 1971.
8. J. H. WILKINSON, "The Algebraic Eigenvalue Problem," pp. 635–636, Oxford Univ. Press (Clarendon), London/New York, 1965.
9. G. PETERS AND J. H. WILKINSON, *SIAM Rev.* **21** (1979), 339–360.
10. G. PETERS AND J. H. WILKINSON, *in* "Handbook for Automatic Computation," Vol. II, "Linear Algebra" (J. H. Wilkinson and C. Reinsch, Eds.), pp. 418–439, Springer–Verlag, Berlin/Heidelberg/New York, 1971.
11. S. C. EISENSTAT, M. C. GURSKY, M. A. SCHULTZ, AND A. H. SHERMAN, "Yale Sparse Matrix Package. I. The Symmetric Codes," Research Report No. 112, Department of Computer Science, Yale University.
12. A. KAHN, J. E. LOWNEY, J. L. BLUE, AND C. L. WILSON, *J. Appl. Phys.* **52** (1981), 4075–4080.