LA

NASA
Technical
Paper
2514

March 1986

Solution of the Symmetric Eigenproblem $AX = \lambda BX$ by Delayed Division

Gaylen A. Thurston
and Nancy J. C. Bains

NASA

NASA
Technical
Paper
2514

1986

Solution of the Symmetric Eigenproblem $AX = \lambda BX$ by Delayed Division

Gaylen A. Thurston
and Nancy J. C. Bains

Langley Research Center
Hampton, Virginia

National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

Contents

Summary

Delayed division is an iterative method for solving the linear eigenproblem $\mathbf{AX} = \lambda\mathbf{BX}$ for a limited number of small eigenvalues and their corresponding eigenvectors. The distinctive feature of the method is the reduction of the problem to an approximate triangular form by systematically dropping quadratic terms in the eigenvalue λ. The report describes the pivoting strategy in the reduction and the method for preserving symmetry in submatrices at each reduction step. Along with the approximate triangular reduction, the report extends some techniques used in the method of inverse subspace iteration. Examples are included for problems of varying complexity.

Introduction

A step in the postbuckling analysis of plates and shells is the computation of several small eigenvalues and corresponding eigenvectors of a linear eigenproblem $\mathbf{AX} = \lambda\mathbf{BX}$. The matrices **A** and **B** that appear in the postbuckling problem are symmetric but not necessarily positive definite. One condition assumed in most methods for solving the eigenproblem is that either matrix **A** or matrix **B** is positive definite, and therefore most standard computer subroutines are not applicable for solving the eigenproblems that arise in the solution of postbuckling problems. A new algorithm for solving the symmetric eigenproblem has been developed as an integral part of postbuckling analysis.

The new algorithm, called "delayed division," solves the symmetric problem $\mathbf{AX} = \lambda\mathbf{BX}$ and is described in this report. The contents of the report are confined to the eigenproblem per se, independent of its application to postbuckling analysis. This report also describes the general properties that make delayed division a suitable choice as a method in the solution of other applied problems as opposed to the choice of the method of determinant search or the method of subspace iteration.

The method of delayed division was first applied to computing roots of lambda matrices (ref. 1). In connection with that problem, two computer subroutines were written for solving the problem $\mathbf{AX} = \lambda\mathbf{BX}$, where **A** and **B** are arbitrary square matrices, for only the eigenvalue nearest to zero and its corresponding eigenvector. One subroutine searched for only real eigenvalues, whereas the second subroutine computed complex eigenvalues. The subroutine for real eigenvalues has been applied by Anderson (ref. 2) to the buckling of periodic lattice structures.

Delayed division is an iterative method. As with other iterative methods for eigenproblems, its rate of convergence and its efficiency are problem dependent. This report contains information that pertains to the efficient application of the method and describes the following: the basic algorithm for delayed division; the logic of a computer subroutine (SPLIT) that was used to implement the algorithm; simple numerical examples that illustrate the computer logic; and example problems that were solved by applying the SPLIT subroutine.

The basic algorithm is an extension of the method of Gaussian elimination for linear algebra problems. Once an eigenvalue is known, Gaussian elimination can be applied to the matrix $[\mathbf{A} - \lambda\mathbf{B}]$ to compute the corresponding eigenvector. The reduction of the matrix to triangular form involves a number of divisions. The method of delayed division is based on a reduction of the matrix $[\mathbf{A} - \lambda\mathbf{B}]$ when the eigenvalue is not known. The divisions in the reduction are approximated by a truncated power series. The approximate divisions are completed after approximate eigenvalues are computed from the truncated triangular form. The completion of the reduction in a later step of the computations is the reason for the name delayed division.

The report describes the logic of the new SPLIT subroutine that was specifically designed for the case in which matrices **A** and **B** are symmetric and the iteration procedure is directed at computing more than one small eigenvalue at a time. As in Gaussian elimination subroutines, the SPLIT subroutine has a forward sweep subroutine and a back substitution subroutine that contain most of the arithmetic operations. The forward sweep reduces matrix **A** to exact triangular form and matrix **B** to approximate triangular form. After computing approximate eigenvalues from the triangular forms, the corresponding eigenvectors are computed in the back substitution. The approximate eigenvalues identified in the forward sweep can be corrected by use of an eigenvalue shift and repetition of the forward sweep and back substitution.

The basic algorithm of the method of delayed division consist of a forward sweep followed by a back substitution or by an eigenvalue shift and can be viewed as an extension of the method of determinant search. In addition to the basic algorithm, an optional strategy in the SPLIT subroutine is a modified shift that corrects more than one eigenvalue and corresponding eigenvector at a time. This option is similar in scope to the method of inverse subspace iteration.

Discarding the assumption that either matrix **A** or matrix **B** is known to be positive definite introduces the need for a pivoting strategy in the solution options. The computer logic for the pivoting strategy and for the modified shift is described in detail in the section entitled "Delayed Division." Each major step in the algorithm for delayed division is contained in a SPLIT subroutine. The subroutines are discussed in the order that they are called by the main subroutine starting with the forward sweep procedure. The numerical operations in each step are illustrated with simple numerical examples.

After the description of the computer logic in the SPLIT subroutine, results are presented from four examples that apply delayed division to the symmetric eigenproblem $\mathbf{AX} = \lambda\mathbf{BX}$. The first example, the finite difference equations for a column buckling problem, provides a measure of the rate of convergence of the iteration as a function of both the number of intervals in the difference equations and the computed number of critical loads. The second example is a textbook problem that demonstrates the numerical stability in the eigenvector computations; the stability results from the pivoting strategy in the forward sweep subroutine. The third example is the shear buckling of simply supported plates, an example which shows the user can often modify problems to speed up convergence. The fourth example is an application of delayed division to lambda matrices derived by the method of Lagrangian multipliers, a practical application that combines the power and the versatility of the method of Langrangian multipliers with the rapid convergence of delayed division. After the examples, the report concludes with a discussion of the similarities and differences between delayed division and other methods for solving the symmetric eigenproblem.

Delayed Division

Basic Algorithm for General Case

The method of delayed division is a simple extension of the fundamental idea in Gaussian elimination. This section reviews the algorithm in reference 1 as applied to the problem $\mathbf{AX} = \lambda\mathbf{BX}$, where **A** and **B** are arbitrary square matrices. This review helps to define the requirements for extending the computer implementation of the method to the symmetric eigenproblem.

The basic algorithm in delayed division follows from the steps in the computation of the eigenvector **X** by the method of Gaussian elimination applied to the equation $(\mathbf{A}-\lambda\mathbf{B})\mathbf{X} = 0$. The elimination procedure requires division by terms of the form $(a + \lambda b)$, where a and b are constants. Division by $(a + \lambda b)$ is approximated for small eigenvalues by multiplication of the two leading terms in the power series for the reciprocal of $(a + \lambda b)$

$$\frac{1}{(a+\lambda b)} = \frac{1}{a} - \frac{\lambda b}{a^2} + \frac{(\lambda b/a)^2}{(a+\lambda b)} \tag{1}$$

$$\frac{1}{(a+\lambda b)} \approx \frac{1}{a} - \frac{\lambda b}{a^2} \qquad \left(\left|\frac{\lambda b}{a}\right| \ll 1\right) \tag{2}$$

The two coefficients $(1/a)$ and $(-b/a^2)$ are stored; thus, division by $(a+\lambda b)$ when λ is unknown can be delayed until a value for λ is assigned.

The approximation in equation (2) is used in reference 1 in a forward sweep procedure to produce an approximate upper triangular matrix. The matrices **A** and **B** are read into auxiliary matrices **C** and **D** such that $\mathbf{C} = \mathbf{A}$ and $\mathbf{D} = -\mathbf{B}$. The forward sweep operates on these auxiliary matrices. A typical operation consists of solving for $\mathbf{X}_k$ in an equation of the form

$$(c_{ik} + \lambda d_{ik})\mathbf{X}_k = -\sum_{j\neq k}(c_{ij} + \lambda d_{ij})\mathbf{X}_j \tag{3a}$$

The exact solution of equation (3a) for $\mathbf{X}_k$ in terms of the $\mathbf{X}_j$ is

$$\mathbf{X}_k = -\sum_{j\neq k}\frac{(c_{ij} + \lambda d_{ij})\mathbf{X}_j}{(c_{ik} + \lambda d_{ik})} \tag{3b}$$

The approximate solution of equation (3a) by use of the right-hand side of equation (2) as the divisor and without quadratic terms in λ in the multiplications is

$$\mathbf{X}_k = -\sum_{j \neq k} (c'_{ij} + \lambda d'_{ij})\mathbf{X}_j \tag{3c}$$

where

$$c'_{ij} = \frac{c_{ij}}{c_{ik}}$$

$$d'_{ij} = \frac{d_{ij}}{c_{ik}} - \frac{c_{ij}d_{ik}}{c_{ik}^2}$$

The terms c'_{ij} and d'_{ij} are stored over c_{ij} and d_{ij}. Equation (3c) is used to eliminate $\mathbf{X}_k$ from the remaining equations; the usual algebraic rules of Gaussian elimination are applied except that quadratic terms in λ are dropped as they appear. An elementary numerical example will show the simplicity of delayed division and, at the same time, will reveal details of numerical analysis that must be kept in mind in the computer implementation. The numerical example is a textbook problem from reference 3, wherein it is used to illustrate convergence of the power iteration for the standard problem $\mathbf{AX} = \lambda\mathbf{X}$. The problem is to find nontrivial solutions of the following homogeneous equations:

$$(3 - \lambda)\mathbf{X}_1 + 2\mathbf{X}_2 + 1\mathbf{X}_3 = 0 \tag{4a}$$

$$2\mathbf{X}_1 + (2 - \lambda)\mathbf{X}_2 + 1\mathbf{X}_3 = 0 \tag{4b}$$

$$1\mathbf{X}_1 + 1\mathbf{X}_2 + (1 - \lambda)\mathbf{X}_3 = 0 \tag{4c}$$

The recursive use of equation (2) follows along the lines of Gaussian elimination, with quadratic terms in λ dropped at each step in the elimination process. Division of equation (4a) by the coefficient $(3 - \lambda)$ is approximated from multiplication by $(1/3) + (\lambda/9)$ to give

$$1\mathbf{X}_1 + [(2/3) + (2\lambda/9)]\mathbf{X}_2 + [(1/3) + (\lambda/9)]\mathbf{X}_3 = 0$$

The next step is elimination of $\mathbf{X}_1$ in equations (4b) and (4c) by multiplying the above equation by the appropriate coefficient of $\mathbf{X}_1$, dropping quadratic terms in λ, and subtracting. The result is

$$1\mathbf{X}_1 + [(2/3) + (2\lambda/9)]\mathbf{X}_2 + [(1/3) + (\lambda/9)]\mathbf{X}_3 = 0 \tag{5a}$$

$$0\mathbf{X}_1 + [(2/3) - (13\lambda/9)]\mathbf{X}_2 + [(1/3) - (2\lambda/9)]\mathbf{X}_3 = 0 \tag{5b}$$

$$0\mathbf{X}_1 + [(1/3) - (2\lambda/9)]\mathbf{X}_2 + [(2/3) - (10\lambda/9)]\mathbf{X}_3 = 0 \tag{5c}$$

Multiplication by $(3/2) + (13\lambda/4)$ is the first step in eliminating $\mathbf{X}_2$ from equation (5c). The resulting equations are

$$1\mathbf{X}_1 + [(2/3) + (2\lambda/9)]\mathbf{X}_2 + [(1/3) + (\lambda/9)]\mathbf{X}_3 = 0 \tag{6a}$$

$$0\mathbf{X}_1 + 1\mathbf{X}_2 + [(1/2) + (3\lambda/4)]\mathbf{X}_3 = 0 \tag{6b}$$

$$0\mathbf{X}_1 + 0\mathbf{X}_2 + [(1/2) + (5\lambda/4)]\mathbf{X}_3 = 0 \tag{6c}$$

For a nontrivial solution, the coefficient of $\mathbf{X}_3$ in equation (6c) must become zero. The coefficient is zero when

$$\lambda = \lambda_1 = 2/5 = 0.40 \tag{7}$$

However, the triangulated equations (6) are not exactly equivalent to equations (4), since quadratic terms in λ have been dropped at each step. The algorithm, through use of equation (2), continues the iteration by application of the following eigenvalue shift μ in equations (4):

$$\lambda = 0.40 + \mu \tag{8}$$

After the shift, equations (4) are

$$(2.6-\mu)\mathbf{X}_1+2\mathbf{X}_2+1\mathbf{X}_3=0 \tag{9a}$$

$$2\mathbf{X}_1+(1.6-\mu)\mathbf{X}_2+1\mathbf{X}_3=0 \tag{9b}$$

$$1\mathbf{X}_1+1\mathbf{X}_2+(0.6-\mu)\mathbf{X}_3=0 \tag{9c}$$

The forward sweep through equations (9) shows the need for pivoting before delayed division can be expected to converge in the general case. After the first elimination step for equations (9), the resulting equations are

$$1\mathbf{X}_1+(0.7692+0.2959\mu)\mathbf{X}_2+(0.3846+0.1479\mu)\mathbf{X}_3=0 \tag{10a}$$

$$0\mathbf{X}_1+(0.0615-1.592\mu)\mathbf{X}_2+(0.2308-0.2959\mu)\mathbf{X}_3=0 \tag{10b}$$

$$0\mathbf{X}_1+(0.2308+0.2959\mu)\mathbf{X}_2+(0.2154-1.148\mu)\mathbf{X}_3=0 \tag{10c}$$

For a given value of μ, the approximation of equation (2) is more accurate for the reciprocal of $(0.2308-0.2959\mu)$ than for the reciprocal of $(0.0615-1.592\mu)$, since the absolute value of b/a is smaller for the first expression. Therefore, a proper choice is to solve for $\mathbf{X}_3$ in equation (10b) and eliminate $\mathbf{X}_3$ in equation (10c). After the complete forward sweep, the equtions are

$$1\mathbf{X}_1+(0.7692+0.2959\mu)\mathbf{X}_2+(0.3846+0.1479\mu)\mathbf{X}_3=0 \tag{11a}$$

$$0\mathbf{X}_1+(0.2667-6.556\mu)\mathbf{X}_2+1\mathbf{X}_3=0 \tag{11b}$$

$$0\mathbf{X}_1+(0.1733+1.422\mu)\mathbf{X}_2+0\mathbf{X}_3=0 \tag{11c}$$

For a nontrivial solution of equations (11),

$$\mu=-\frac{0.1733}{1.422}=-0.121875 \tag{12}$$

$$\lambda=0.40+\mu=0.278125 \tag{13}$$

For this value of μ, the approximate reciprocal of $(0.2308-0.2959\mu)$ calculated from equation (2) is correct to within 3 percent, whereas the approximation for $(0.0615-1.592\mu)$ has the wrong algebraic sign, since $|\mathrm{b}\mu/a|>1$. Therefore, the use of the approximate reciprocal from equation (2) as part of a numerically stable forward sweep algorithm requires permuting the order of elimination of the variables, or "pivoting for size." (See ref. 4.) The algorithms used in the subroutines described in reference 1 use "maximal pivots" determined by selecting the maximum absolute value of c_{ij} from the coefficients $(c_{ij}+\lambda d_{ij})$ that remain in the equations of lower order after each elimination step. Maximal pivoting on the off-diagonal terms destroys the symmetry of the reduced system of equations of lower order when **A** and **B** are symmetric matrices. Preservation of symmetry to decrease the total number of calculations is the motivation for using a different pivoting strategy for symmetric problems.

Before another pivoting strategy is presented, the convergence of the iteration calculated with the approximate reciprocal and an eigenvalue shift is summarized below.

$$\begin{array}{ll}
\lambda_1^{(1)}=0.40 & \lambda_1^{(4)}=0.3079788 \\
\lambda_1^{(2)}=0.278125 & \lambda_1^{(5)}=0.3079785 \\
\lambda_1^{(3)}=0.3085412 & \text{Exact } \lambda_1=0.30797853
\end{array}$$

Note that the eigenvector associated with λ can be computed by back substitution in the triangulated equations when μ, the correction on λ, is small.

Approximate 2 × 2 Matrix Inverse

The pivoting strategy in the algorithm in reference 1 destroys the symmetry of the reduced system of equations of lower order when an off-diagonal term is the pivot element in an elimination step. This section introduces another pivoting strategy that preserves the symmetry of the original problem. The new pivoting strategy is based on eliminating two variables at a time while retaining a systematic procedure for dropping quadratic terms in λ as they appear. Eliminating two variables at a time involves computation and storage of an approximate inverse of a symmetric 2×2 matrix. If the matrix is written as

$$(\mathbf{C}+\lambda\mathbf{D}) = \begin{bmatrix} (c_{ii}+\lambda d_{ii}) & (c_{ij}+\lambda d_{ij}) \\ (c_{ij}+\lambda d_{ij}) & (c_{jj}+\lambda d_{jj}) \end{bmatrix} \tag{14}$$

the exact inverse is

$$(\mathbf{C}+\lambda\mathbf{D})^{-1} = \begin{bmatrix} (c_{jj}+\lambda d_{jj})/\Delta & -(c_{ij}+\lambda d_{ij})/\Delta \\ -(c_{ij}+\lambda d_{ij})/\Delta & (c_{ii}+\lambda d_{ii})/\Delta \end{bmatrix} \tag{15}$$

which is also symmetric. In equation (15), the divisor Δ is the determinant of $(\mathbf{C}+\lambda\mathbf{D})$:

$$\Delta = a + b\lambda + c\lambda^2 \tag{16}$$

where $a = (c_{ii}c_{jj} - c_{ij}^2)$, $b = (c_{ii}d_{jj} + c_{jj}d_{ii} - 2c_{ij}d_{ij})$, and $c = (d_{ii}d_{jj} - d_{ij}^2)$. The approximate inverse used in delayed division is

$$(\mathbf{C}+\lambda\mathbf{D})^{-1} = \begin{bmatrix} (c'_{ii}+\lambda d'_{ii}) & (c'_{ij}+\lambda d'_{ij}) \\ (c'_{ij}+\lambda d'_{ij}) & (c'_{jj}+\lambda d'_{jj}) \end{bmatrix} \tag{17}$$

where

$$c'_{ii} = \frac{c_{jj}}{a} \quad d'_{ii} = \frac{d_{jj}}{a} - \frac{bc_{jj}}{a^2}$$

$$c'_{ij} = \frac{-c_{ij}}{a} \quad d'_{ij} = \frac{-d_{ij}}{a} + \frac{bc_{ij}}{a^2}$$

$$c'_{jj} = \frac{c_{ii}}{a} \quad d'_{jj} = \frac{d_{ii}}{a} - \frac{bc_{ii}}{a^2}$$

The approximation in equation (17) follows from dropping the quadratic term in λ in equation (16) and then applying the approximation of equation (2) to the division by Δ in equation (15). The approximation preserves the symmetry of the inverse. The approximate inverse in equation (17) is the unique feature of delayed division for the symmetric problem $\mathbf{AX} = \lambda\mathbf{BX}$. Before the approximation in equation (17) is applied to the general elimination procedure, it will be used on the simple numerical example in equations (4). The leading 2×2 coefficient matrix in equations (4) (e.g., $\mathbf{A}_2$) corresponds to equation (14) as follows:

$$\mathbf{A}_2^{-1} = \begin{bmatrix} (3-\lambda) & 2 \\ 2 & (2-\lambda) \end{bmatrix}^{-1} = \begin{bmatrix} (2-\lambda)/\Delta & -2/\Delta \\ -2/\Delta & (3-\lambda)/\Delta \end{bmatrix} \tag{18}$$

where $\Delta = (3-\lambda)(2-\lambda) - 4 = 2 - 5\lambda + \lambda^2$. The approximate inverse of $\mathbf{A}_2$ is

$$\mathbf{A}_2^{-1} = \begin{bmatrix} (1+2\lambda) & -1-(5\lambda/2) \\ -1-(5\lambda/2) & (3/2)+(13\lambda/4) \end{bmatrix} \tag{19}$$

Equation (19) follows from equation (18) with $(1/\Delta)$ approximated by $(1/2) + (5\lambda/4)$ and the quadratic terms in λ dropped. After multiplication of equations (4a) and (4b) by $\mathbf{A}_2^{-1}$, elimination of $\mathbf{X}_1$ and $\mathbf{X}_2$ from equation (4c), and elimination of quadratic terms in λ at each step, the resulting equations are

$$1\mathbf{X}_1 + 0\mathbf{X}_2 + (0 - \lambda/2)\mathbf{X}_3 = 0 \tag{20a}$$

$$0\mathbf{X}_1 + 1\mathbf{X}_2 + [(1/2) + (3\lambda/4)]\mathbf{X}_3 = 0 \tag{20b}$$

$$0\mathbf{X}_1 + 0\mathbf{X}_2 + [(1/2) - (5\lambda/4)]\mathbf{X}_3 = 0 \tag{20c}$$

Equations (20) and equations (6) are equivalent; they have the same nontrivial solution. The order in which quadratic terms are dropped does not affect the final result of the forward sweep.

At the next iteration step for the numerical example, the pivoting problem that arises in solving equations (9) is resolved by first solving the equations

$$(2.6 - \mu)\mathbf{X}_1 + 1\mathbf{X}_3 = -2\mathbf{X}_2 \tag{21a}$$

$$1\mathbf{X}_1 + (0.6 - \mu)\mathbf{X}_3 = -1\mathbf{X}_2 \tag{21b}$$

for $\mathbf{X}_1$ and $\mathbf{X}_3$ in terms of $\mathbf{X}_2$. The choice of order of elimination is based on the determinant for the coefficients on the left side of equations (21). That determinant is $\Delta = 0.56 - 3.2\Delta\mu + \Delta\mu^2$, whereas the determinant for the leading 2×2 coefficient matrix of equations (9) is $\Delta = (2.6 - \Delta\mu)(1.6 - \Delta\mu) - 4 = 0.16 - 4.2\Delta\mu + \Delta\mu^2$. The constant term 0.56 is larger than 0.16, so that $\mathbf{X}_1$ and $\mathbf{X}_3$ are eliminated first with equations (21). Again, the final triangular equations at the end of the forward sweep are equivalent. The symmetry of the reduced set of equations at each step of the forward sweep is not readily apparent for the simple 3×3 matrix here. The preservation of symmetry is illustrated in the next subsection.

Forward Sweep

In this section, three numerical operations from the forward sweep procedure for the symmetric problem are outlined in more detail. The first operation is the use of 2×2 inverse matrices in the elimination procedure. Symmetry in the reduced equations is preserved for the general case. The second operation is the pivoting strategy, that is, the choice of the order in which variables are eliminated in the SPLIT subroutine. By knowing the pivoting strategy, the subroutine user can often scale applied problems to permute the order that variables are eliminated. The effect of scaling problems with equivalent matrices is discussed briefly. The third operation is inclusion of quadratic terms in the determinant of the final 2×2 matrix at the end of the sweep. The third operation gives two approximate small eigenvalues rather than one, with no additional approximations.

The first operation considered herein is the elimination of two variables at a time in the forward sweep procedure. As in Gaussian elimination for linear algebraic equations, the $\mathbf{A}$ matrix is replaced by a sequence of equivalent matrices. In the computer algorithm, only the current equivalent matrix is stored. The notation adopted here is to let $\mathbf{C} = \mathbf{A}$ and $\mathbf{D} = -\mathbf{B}$ at the start of the forward sweep. Before each elimination step, the equations are of the form

$$(\mathbf{C} + \lambda\mathbf{D})\mathbf{X} = 0 \tag{22}$$

After the elimination step, a prime is used to distinquish the new set of equations:

$$(\mathbf{C}' + \lambda\mathbf{D}')\mathbf{X} = 0 \tag{23}$$

The terminology "reduced set of equations" is usually applied to the new set of equations (eqs. (23)). However, in this report "reduced equations" are the equations in variables that remain to be "eliminated" in equations (23). To condense the notation somewhat, let

$$g_{ij} = c_{ij} + \lambda d_{ij} \qquad \left\{ \begin{array}{l} (i = 1, 2, ..., n) \\ (j = 1, 2, ..., n) \end{array} \right.$$

where c_{ij} and d_{ij} are elements of the $n \times n$ matrices $\mathbf{C}$ and $\mathbf{D}$.

At the first elimination step, equations (22) are symmetric so that $g_{ji} = g_{ij}$. This symmetry is preserved in the $n - 2$ reduced equations after the first step, $g'_{ji} = g'_{ij}$. The elimination process is then repeated for the reduced set. The symmetry of the reduced equations is apparent when the elimination step is written in detail. The equations, before the first elimination step, are of the form

$$\sum_{j=1}^{n} g_{ij}x_j = 0 \qquad (i = 1, 2, ..., n) \tag{24}$$

The operation considered is the elimination of $\mathbf{X}_I$ and $\mathbf{X}_J$ where I and J are selected so that

$$\left|\mathbf{C}_{II}\mathbf{C}_{JJ} - \mathbf{C}_{IJ}^2\right| \geq \left|c_{ii}c_{jj} - c_{ij}^2\right| \qquad \left\{ \begin{array}{l} (i = 1, 2, ..., n) \\ (j = 1, 2, ..., n) \end{array} \right.$$

In other words, I and J are selected to give the maximum absolute value for the symmetric 2×2 submatrices that appear in the coefficient matrix for the equations. The Ith and Jth equations of equations (24) are written

$$g_{II}x_I + g_{IJ}x_J = -\sum_{j \neq I,J}^{n} g_{Ij}x_j \tag{25a}$$

$$g_{IJ}x_I + g_{JJ}x_J = -\sum_{j \neq I,J}^{n} g_{Jj}x_j \tag{25b}$$

The exact inverse matrix of coefficients on the left side of equations (25) is defined in a manner similar to equation (15) as

$$\mathbf{G}_{IJ}^{-1} = \begin{bmatrix} g_{II}^{-1} & g_{IJ}^{-1} \\ g_{IJ}^{-1} & g_{JJ}^{-1} \end{bmatrix} \tag{26}$$

with

$$g_{II}^{-1} = (c_{JJ} + \lambda d_{JJ})/\Delta \tag{27a}$$

$$g_{IJ}^{-1} = -(c_{IJ} + \lambda d_{IJ})/\Delta \tag{27b}$$

$$g_{JJ}^{-1} = (c_{II} + \lambda d_{II})/\Delta \tag{27c}$$

where, as in equation (16),

$$\Delta = a + b\lambda + c\lambda^2$$

and $a = (c_{II}d_{JJ} - c_{IJ}^2)$, $b = (c_{II}d_{JJ} + c_{JJ}d_{II} - 2c_{IJ}d_{IJ})$, and $c = (d_{II}d_{JJ} - d_{IJ}^2)$. Multiplying equations (25) by the inverse matrix of equations (26) yields

$$x_I = -\sum_{j \neq I,J} g'_{Ij}x_j \tag{28a}$$

$$x_J = -\sum_{j \neq I,J} g'_{Jj}x_j \tag{28b}$$

where

$$g'_{Ij} = g_{II}^{-1}g_{Ij} + g_{IJ}^{-1}g_{Jj}$$

$$g'_{Jj} = g_{IJ}^{-1}g_{Ij} + g_{JJ}^{-1}g_{Jj}$$

When $i \neq I$ or $i \neq J$ in equations (24), the typical equation with the row subscript i is

$$g_{iI}x_I + g_{iJ}x_J + \sum_{j \neq I,J} g_{ij}x_j = 0 \qquad (i \neq I,\ i \neq J) \tag{29}$$

The variables x_I and x_J are eliminated from equations (29) by substitution of the right side of equations (28) to give

$$\sum_{j \neq I,J} g'_{ij}x_j = 0 \qquad (i \neq I,\ i \neq J) \tag{30}$$

where

$$g'_{ij} = g_{ij} - g_{iI}g'_{Ij} - g_{iJ}g'_{Jj} \tag{31}$$

or

$$g'_{ij} = g_{ij} - g_{iI}(g_{II}^{-1}g_{Ij} + g_{IJ}^{-1}g_{Jj}) - g_{iJ}(g_{IJ}^{-1}g_{Ij} + g_{JJ}^{-1}g_{Jj}) \tag{32}$$

In equation (32), interchanging subscripts shows directly that

$$g'_{ji} = g'_{ij} \tag{33}$$

Therefore, the reduced equations (30) are symmetric. In the computer subroutine, the reduced-equation coefficients are truncated to linear terms in λ and then the c'_{ij} and d'_{ij} terms are stored over the c_{ij} and d_{ij} coefficients. Each term in equation (32) is a product or quotient of elementary linear factors of the form $(c+\lambda d)$ so that the truncation to linear terms in λ can follow any order. The arbitrary order of truncation makes it convenient to truncate equations (27) first. The coefficients in the term

$$g_{II}^{-1} = \left(\frac{c_{JJ}}{a}\right) + \lambda\left(\frac{d_{JJ}}{a} - \frac{c_{JJ}b}{a^2}\right) \tag{34}$$

are stored in the original locations for c_{II} and d_{II} and the coefficients of g_{IJ}^{-1} and g_{JJ}^{-1} are stored in a similar manner.

The purpose of preserving symmetry in the reduced equations is to decrease the number of operations in the numerical solution. Only half the off-diagonal terms are computed at each elimination step.

The second operation in the forward sweep that is considered in more detail herein is the pivoting strategy. The variables x_I and x_J to be eliminated at each step are arbitrarily chosen so that the coefficient $a = (c_{II}c_{JJ} - c_{IJ}^2)$ has a maximum absolute value of all possible $(c_{ii}c_{jj} - c_{ij}^2)$. The accuracy of the approximate truncation to linear terms of the inverse matrix G_{IJ}^{-1} defined by equations (26) and (27) depends on an application of equation (7). The reciprocal for Δ, defined in equation (16), is

$$\frac{1}{\Delta} = \frac{1}{a}\left[1 - \lambda\left(\frac{b}{a} + \frac{\lambda c}{a}\right)\right] + \lambda^2\frac{[(b/a) + (\lambda c/a)]^2}{\Delta} \tag{35}$$

For the leading term in equation (35) to approximate the reciprocal of Δ, the term $(\lambda b/a) + (\lambda^2 c/a)$ must satisfy the inequality

$$\left|\lambda\left(\frac{b}{a} + \frac{\lambda c}{a}\right)\right| < 1 \tag{36}$$

The accuracy of the truncated expression

$$\frac{1}{\Delta} = \frac{1}{a}\left(1 - \frac{\lambda b}{a}\right) \tag{37}$$

depends on λ, b, and c as well as the constant a. However, the algorithm programmed in the SPLIT subroutine only tests for the magnitude of a. Since λ is not known, there is no control over λ during the forward sweep. Before the sweep procedure starts, an eigenvalue shift can be used to speed up convergence.

The ratios (b/a) and (c/a) appearing in inequality (36) could be used as a basis for a test to select pivot variables x_I and x_J. Such a test would depend on matrix elements d_{ij} in addition to c_{ij} and would require more computer operations per elimination step, so that this additional testing is neither explored further herein nor adopted in the SPLIT subroutine. However, the subroutine user can exert some control over the sweep process by scaling the original problem. One means of scaling the symmetric problem $(\mathbf{A} - \lambda\mathbf{B})\mathbf{X} = 0$ is to change $\mathbf{X}$ to $\mathbf{RY}$ and multiply the matrix equation by $\mathbf{R}^T$, the transpose of $\mathbf{R}$. The original symmetric problem becomes a symmetric equivalent problem

$$(\mathbf{R}^T\mathbf{AR})\mathbf{Y} = \lambda(\mathbf{R}^T\mathbf{BR})\mathbf{Y} \tag{38}$$

with the same eigenvalues. The matrix $\mathbf{C}$ at the start of the forward sweep is $\mathbf{R}^T\mathbf{AR}$. The order of elimination of variables for the matrix $\mathbf{C}$ in the subroutine can obviously be different from that of the original matrix $\mathbf{A}$.

Fortunately, most mechanics problems tend to scale naturally. Scaling is illustrated in this report for the problem which applies the method of Lagrangian multipliers to plate buckling.

The pivoting strategy adopted herein is numerically stable and the elimination procedure is exact when λ is zero. The operations in which matrix $\mathbf{A}$ is factored contain no approximations. If the interchange of row and column matrix elements is included in the forward sweep procedure, the final matrix $\mathbf{C}$ is triangular and the product of the $(c_{II}c_{JJ} - c_{IJ}^2)$ terms generated by the sweep are equal to the determinant of $\mathbf{A}$. Interchanges of elements are not included in the forward sweep procedure in the SPLIT subroutine, but the value of the determinant is independent of interchanges and is computed as a product after the sweep.

The third operation in the forward sweep occurs at the last elimination of the sweep, when the reduced set of equations contains two equations. In this case, the eigenvalues of the determinant of the last 2 × 2 reduced matrix are computed exactly. The computation gives two eigenvalues that are approximations of small eigenvalues of the original problem.

This last operation is not illustrated in the elementary numerical example considered previously; the 3 × 3 matrix of that example is too small to show the practical significance. The following augmented problem is a better example:

$$(3-\lambda)\mathbf{X}_1 + 2\mathbf{X}_2 + 1\mathbf{X}_3 + 0\mathbf{X}_4 = 0 \tag{39a}$$

$$2\mathbf{X}_1 + (2-\lambda)\mathbf{X}_2 + 1\mathbf{X}_3 + 0\mathbf{X}_4 = 0 \tag{39b}$$

$$1\mathbf{X}_1 + 1\mathbf{X}_2 + (1-\lambda)\mathbf{X}_3 + 0\mathbf{X}_4 = 0 \tag{39c}$$

$$0\mathbf{X}_1 + 0\mathbf{X}_2 + 0\mathbf{X}_3 + (4-\lambda)\mathbf{X}_4 = 0 \tag{39d}$$

The numerical example defined by equations (39) has eigenvalue $\lambda_4 = 4$ plus the eigenvalues of the first numerical example.

The forward sweep procedure starts with $I = 1$ and $J = 4$. To eliminate $\mathbf{X}_1$ and $\mathbf{X}_4$, the inverse matrix (eqs. (26)) is

$$\mathbf{G}_{14}^{-1} = \begin{bmatrix} \frac{(4-\lambda)}{12-7\lambda+\lambda^2} & (0+0\lambda) \\ (0+0\lambda) & \frac{(3-\lambda)}{12-7\lambda+\lambda^2} \end{bmatrix} \approx \begin{bmatrix} (1/3)+(\lambda/9) & (0+0\lambda) \\ (0+0\lambda) & (1/4)+(\lambda/16) \end{bmatrix} \tag{40}$$

After the elimination of $\mathbf{X}_1$ and $\mathbf{X}_4$ in the reduced equations, equations (39) become

$$1\mathbf{X}_1 + [(2/3)+(2\lambda/9)]\mathbf{X}_2 + [(1/3)+(\lambda/9)]\mathbf{X}_3 + 0\mathbf{X}_4 = 0 \tag{41a}$$

$$0\mathbf{X}_1 + [(2/3)-(13\lambda/9)]\mathbf{X}_2 + [(1/3)-(2\lambda/9)]\mathbf{X}_3 + 0\mathbf{X}_4 = 0 \tag{41b}$$

$$0\mathbf{X}_1 + [(1/3)-(2\lambda/9)]\mathbf{X}_2 + [(2/3)-(10\lambda/9)]\mathbf{X}_3 + 0\mathbf{X}_4 = 0 \tag{41c}$$

$$0\mathbf{X}_1 + 0\mathbf{X}_2 + 0\mathbf{X}_3 + 1\mathbf{X}_4 = 0 \tag{41d}$$

The reduced equations in $\mathbf{X}_2$ and $\mathbf{X}_3$ are equations (41b) and (41c), which are symmetric. For a nontrivial solution, the determinant Δ of the coefficients must vanish:

$$\Delta = [(2/3)-(13\lambda/9)][(2/3)-(10\lambda/9)] - [(1/3)-(2\lambda/9)]^2 = 0 \tag{42}$$

The roots of equation (42) are

$$\lambda_1 = \left(7-\sqrt{7}\right)/14 = 0.3110 \tag{43a}$$

$$\lambda_2 = \left(7+\sqrt{7}\right)/14 = 0.6890 \tag{43b}$$

The exact eigenvalues of the given problem (eqs. (39)) are $\lambda_1 = 0.30797853$, $\lambda_2 = 0.64310413$, $\lambda_3 = 5.04891734$, and $\lambda_4 = 4$. The roots of equation (42) approximate the first two eigenvalues of the problem. In particular, the approximation for $\lambda_1 = 0.3110$ is much better than the value $\lambda_1 = 0.40$ obtained from equations (6) for the first numerical example. Equations (6) follow from dropping linear terms in λ in the forward sweep until only one reduced equation remains. For the second numerical example, there are two small eigenvalues and two larger ones and delayed division provides a good approximation to the first two since the approximate inverse matrix in equations (40) is accurate for $\lambda = \lambda_1$ and $\lambda = \lambda_2$.

In the general case, the determinant (eq. (16)) of the final 2 × 2 set of equations is of the form

$$\Delta = a + b\lambda + c\lambda^2$$

with roots

$$\lambda = \frac{-b \pm (b^2 - 4ac)^{1/2}}{2c} \tag{44}$$

As noted in the discussion on the pivoting strategy for selecting I and J in equations (25), the constant a is an exact factor of the determinant of the matrix $\mathbf{A}$. Therefore, if $a = 0$, the root $\lambda = 0$ of equation (16) is an exact eigenvalue of $(\mathbf{A} - \lambda\mathbf{B})\mathbf{X} = 0$. When $a \neq 0$ and the roots of equation (44) are real and small, the roots provide approximations to two real eigenvalues for $(\mathbf{A} - \lambda\mathbf{B})X = 0$.

When the discriminant $(b^2 - 4ac)$ is negative, the roots of equation (44) are complex numbers and the SPLIT subroutine returns an error message. This raises the question of whether the original problem has complex eigenvalues. An eigenvalue shift to the real part of the complex conjugate eigenvalues allows a more precise check for complex eigenvalues, or the complex algorithm of reference 1 can be applied.

Computation of the roots of the determinant of the last pair of reduced equations with equation (44) completes the forward sweep procedure. Use of approximate 2 × 2 inverse matrices reduces the number of computations needed in the sweep because of symmetry. Complete pivoting provides a stable computation of the determinant of the matrix $\mathbf{A}$. The pivoting also provides a systematic method of dropping quadratic terms in λ, although this strategy may not be optimum.

The two roots of equation (44), λ_1 and λ_2, are small approximate roots of the full eigenproblem. Approximate eigenvectors can be computed by back substitution similar to back substitution in Gaussian elimination.

Back Substitution

This section contains the back substitution procedure for the second numerical example and summarizes the matrix notation for the forward sweep and for the back substitution. Two approximate eigenvectors are computed by back substitution for the second numerical example. The general case allows for m approximate eigenvectors to be computed. The m eigenvectors correspond to m approximate eigenvalues identified in the forward sweep. The method of identification is outlined in this section.

Some additional notation is introduced in this section. The approximate eigenvector corresponding to the approximate eigenvalue λ_k is denoted as $\mathbf{V}_k$. The components of $\mathbf{V}_k$ are v_{kj}:

$$\mathbf{V}_k^T = (v_{k1}, v_{k2}, ..., v_{kj}) \qquad (j = 1, 2, ..., n) \tag{45}$$

The notation $\mathbf{U}_k$ is reserved for the exact eigenvectors that satisfy the original problem:

$$\mathbf{A}\mathbf{U}_k = \lambda_k \mathbf{B}\mathbf{U}_k \tag{46}$$

The second numerical example is defined by equations (39). At the end of the forward sweep procedure, the approximate equations are equations (41). Equations (41b) and (41c) are decoupled equations in $\mathbf{X}_2$ and $\mathbf{X}_3$. For a nontrivial solution, the two equations must be linearly dependent. The linearly dependent equations for the components of the eigenvector corresponding to $\lambda_1 = 0.3110$ are written as

$$\begin{aligned} v_{13} &= \frac{-[(2/3) - (13\lambda_1/9)]}{[(1/3) - (2\lambda_1/9)]} v_{12} = \frac{-[(1/3) - (2\lambda_1/9)]}{[(2/3) - (10\lambda_1/9)]} v_{12} \\ v_{13} &= -0.8229 v_{12} \end{aligned} \tag{47}$$

The arbitrary constant in the eigenvector is fixed by setting $v_{12} = 1$. The remaining two equations (41) determine v_{11} and v_{14} as follows:

$$1v_{11} + 0v_{14} = -\{[(2/3) + (2\lambda_1/9)]v_{12} + [(1/3) + (\lambda_1/9)]v_{13}\} \tag{48a}$$

$$0v_{11} + 1v_{14} = -(0v_{12} + 0v_{13}) \tag{48b}$$

Substitution of λ_1, v_{12}, and v_{13} into equations (48) leads directly to $v_{11} = -0.43305$ and $v_{14} = 0$.

The first eigenvector of equations (41) corresponding to $\lambda_1 = 0.3110$ is, therefore,

$$\mathbf{V}_1^T = (-0.4331, \ 1.0, \ -0.8229, \ 0) \tag{49}$$

The second eigenvector for $\lambda_2 = 0.6890$ is

$$\mathbf{V}_2^T = (0.8596,\ 0.5486,\ 1.0,\ 0) \tag{50}$$

The computations for the back substitution in the method of delayed division are straightforward. The implicit division that is approximated in the forward sweep is completed by explicit substitution of λ_k in the back substitution.

The algorithm for improving the approximation of $m = 2$ eigenvalues and eigenvectors for the second numerical example by the use of $\mathbf{V}_1$ and $\mathbf{V}_2$ is completed in the next section. Improving the approximation for the general case of m eigenvalues, where $m > 2$, is discussed further here.

The procedure for selecting $m > 2$ approximate eigenvalues is part of the forward sweep procedure. For each set of equations in the form of equations (25), two eigenvalues are computed from the matrix of coefficients on the left-hand side. The matrix of coefficients is

$$\mathbf{G}_{IJ} = \begin{bmatrix} g_{II} & g_{IJ} \\ g_{IJ} & g_{JJ} \end{bmatrix} = \begin{bmatrix} (c_{II} + \lambda d_{II}) & (c_{IJ} + \lambda d_{IJ}) \\ (c_{IJ} + \lambda d_{IJ}) & (c_{JJ} + \lambda d_{JJ}) \end{bmatrix} \tag{51}$$

The determinant Δ of $\mathbf{G}_{IJ}$ has two eigenvalues determined by the roots of

$$\Delta = (c_{II} + \lambda d_{II})(c_{JJ} + \lambda d_{JJ}) - (c_{IJ} + \lambda d_{IJ})^2 = 0 \tag{52}$$

The eigenvalues in equations of the form of equation (52) are numbered in reverse order of their computation. That is, on the rth elimination step the two roots of the 2 × 2 matrix determinant are stored as λ_{n+2-2r} and λ_{n+1-2r}. At the end of the forward sweep, the m approximate eigenvalues are selected from the λ_i with the m lowest subscripts.

The first step in improving the m eigenvalues is the computation of approximate eigenvectors from back substitution. The order of the operations in the back substitution is the reverse of the elimination order in the forward sweep. In the computer subroutine, a permutation vector $\mathbf{P}$ with components $\mathbf{P}_k$ is used to record the order of operations. When $\mathbf{X}_I$ and $\mathbf{X}_J$ are the rth pair of unknowns eliminated in the forward sweep, two components of the permutation vector are defined as $P_{2r-1} = I$ and $P_{2r} = J$. The permutation vector determined by the pivoting strategy for the second numerical example (eqs. (39)) is

$$\mathbf{P}^T = (1, 4, 2, 3) \tag{53}$$

The permutation vector is used in the back substitution, but substitution is simpler for the case in which the variables are eliminated in their natural order of $P_i = i$. For this special case, let $k = n + 1 - 2r$. Then, for a nontrivial solution of equations (25) when $\lambda = \lambda_k$, the components of the approximate eigenvector $\mathbf{V}_k$ obey the relations

$$\begin{aligned} v_{kI} &= \frac{-[(c_{IJ} + \lambda_k d_{IJ}) v_{kJ} + \sum_{j=2r+1}^{n} g_{Ij} v_{kJ}]}{(c_{II} + \lambda_k d_{JJ})} \\ &= \frac{-[(c_{JJ} + \lambda_k d_{JJ}) v_{kJ} + \sum_{j=2r+1}^{n} g_{Jj} v_{kj}]}{(c_{IJ} + \lambda_k d_{IJ})} \end{aligned} \tag{54}$$

where $I = 2r - 1$ and $J = 2r$. The information necessary to compute v_{kj} for $j > 2r$ is contained in quadratic and higher terms in λ that have been dropped. Therefore, the components v_{kj} for $j > 2r$ are set equal to zero at this stage of the iteration. Either v_{kI} or v_{kJ} is set equal to one and equations (54) define v_{kJ} or v_{kI} with absolute value less than one. The remaining v_{kj} for $j \leq 2r - 2$ follow by direct back substitution in the triangular equations formed during the forward sweep.

At the end of the forward sweep and the back substitution, m approximate eigenvalues and eigenvectors have been identified. The accuracy of the approximation is problem dependent; however, the smaller eigenvalues in absolute value are more accurate. One optional solution strategy to improve the approximations is to shift to a given approximate eigenvalue and repeat the forward sweep and the back substitution. A second optional solution is to assume that the m eigenvectors are useful approximations that can be improved without repeating all the operations in the forward sweep procedure.

The two solutions are indicated in a schematic diagram in figure 1. The iteration in the first solution makes an outer loop that repeats the forward sweep procedure. The optional second solution contains an additional inner loop that applies a modified eigenvalue shift. The steps in the inner loop are discussed next.

Equivalence Transformation

The forward sweep and back substitution procedures identify m approximate small eigenvalues and their associated eigenvectors. This section outlines an equivalence transformation that improves the approximations. In addition to improving the approximate eigenvalues, the equivalence transformation allows a modified forward sweep on the transformed problem. The modified forward sweep and the back substitution require fewer operations than the full forward sweep and the back substitution. The modified forward sweep and back substitution procedures complete the first iteration cycle. The modified sweep procedures are described in the next section.

The equivalence transformation for the original problem

$$\mathbf{AX} = \lambda \mathbf{BX} \tag{55}$$

consists of a change of variables

$$\mathbf{X} = \mathbf{QY} \tag{56}$$

Followed by multiplication by $\mathbf{Q}^T$. The transformed problem is thus

$$(\mathbf{Q}^T\mathbf{AQ})\mathbf{Y} = \lambda(\mathbf{Q}^T\mathbf{BQ})\mathbf{Y} \tag{57}$$

The original problem and the transformed problem are called equivalent because they have the same eigenvalues. A completely general equivalence tansformation allows premultiplication by any conformable matrix. The choice of $\mathbf{Q}^T$ for premultiplication makes the transposed problem symmetric, since $\mathbf{A} = \mathbf{A}^T$ and $\mathbf{B} = \mathbf{B}^T$.

The reason for transforming the problem is to make it simpler in some sense. The canonical form of the change of variables is the special case of $\mathbf{Q} = \mathbf{U}$, where the columns of $\mathbf{U}$ are the normalized eigenvectors $\mathbf{U}_k$ of the original problem. The normalized eigenvectors obey the orthogonality relation

$$\mathbf{U}_k^T\mathbf{BU}_j = \delta_{kj} \tag{58}$$

where $\delta_{kj} = 0$ if $k \neq j$ and $\delta_{kk} = 1$. The choice of $\mathbf{Q} = \mathbf{U}$ makes the transformed problem (eqs. (57))

$$\mathbf{\Lambda Y} = \lambda \mathbf{IY} \tag{59}$$

The matrix $\mathbf{\Lambda} = \mathbf{U}^T\mathbf{AU}$ is a diagonal matrix with elements equal to the eigenvalues λ_k, since the normalized eigenvectors are scaled so that $\mathbf{U}^T\mathbf{BU} = \mathbf{I}$.

The canonical (eqs. (59)) is the end result when all n eigenvalues and eigenvectors of equations (55) have been determined. In the method of delayed division, $m < n$ eigenvectors are computed at one time. Transformed equations with m exact solutions are examined next to provide insight into the procedure for improving m approximate solutions.

Consider the transformation when the last m columns of $\mathbf{Q}$ are eigenvectors $\mathbf{U}_k$ and the remaining elements are zeroes except for the remaining diagonal elements, which are unity. In partitioned form, $\mathbf{Q}$ is written

$$\mathbf{Q} = \left[\begin{array}{c|c} \mathbf{I}_{n-m,n-m} & \\ \hline & \mathbf{U}_{n,m} \\ \mathbf{0}_{m,n-m} & \end{array}\right] \tag{60}$$

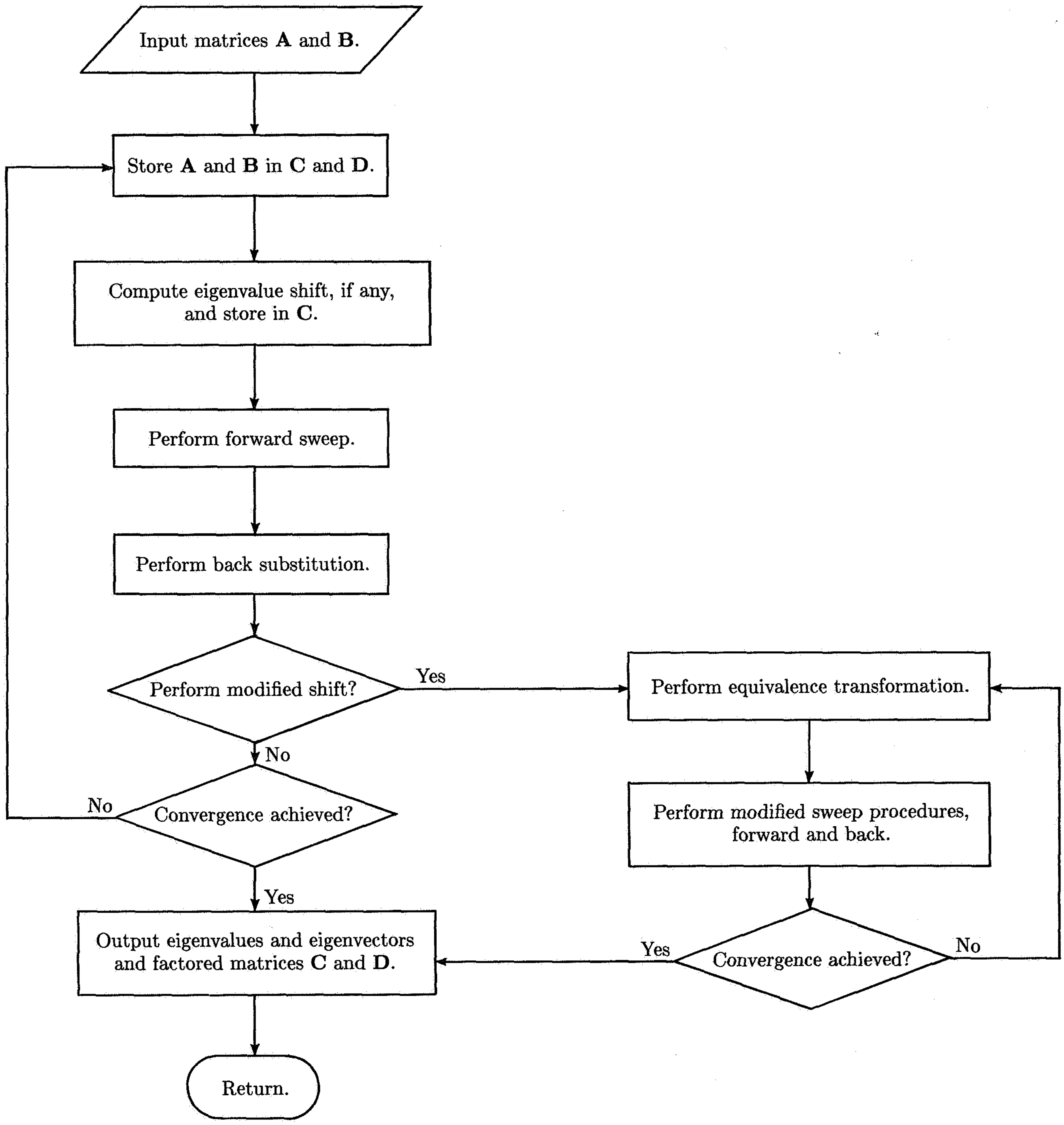
Input matrices **A** and **B**.
Store **A** and **B** in **C** and **D**.
Compute eigenvalue shift, if any, and store in **C**.
Perform forward sweep.
Perform back substitution.
Perform modified shift?
Yes
No
Convergence achieved?
No
Yes
Perform equivalence transformation.
Perform modified sweep procedures, forward and back.
Convergence achieved?
Yes
No
Output eigenvalues and eigenvectors and factored matrices **C** and **D**.
Return.

Figure 1. Schematic diagram of procedure for solving symmetric eigenvalue problem $\mathbf{AX} = \lambda\mathbf{BX}$ by delayed division.

where the subscripts on the submatrices denote the number of rows and columns. It is useful here to consider the computational steps of the transformation in detail since the computer algorithm for the equivalence transformation follows the same steps. The matrix products (**AQ**) and (**BQ**) have the partitioned forms

$$(\mathbf{AQ}) = \left[\begin{array}{c|c} \mathbf{A}_{n,n-m} & (\mathbf{AU})_{n,m} \end{array}\right] \tag{61a}$$

and

$$(\mathbf{BQ}) = \left[\begin{array}{c|c} \mathbf{B}_{n,n-m} & (\mathbf{BU})_{n,m} \end{array}\right] \tag{61b}$$

The notation $\mathbf{A}_{n,n-m}$ and $\mathbf{B}_{n,n-m}$ is used to indicate that the first $(n-m)$ columns of the matrices **A** and **B** are unchanged by the multiplication. Each column k of $(\mathbf{AU})_{n,m}$ is of the form

$$\mathbf{AU}_k = \lambda_k \mathbf{BU}_k \tag{62}$$

If $\mathbf{U}_k$ is in column $(n+1-k)$ of **Q**, $(\mathbf{AU}_k)$ is in the same column of (**AQ**). Multiplication of equations (61) by $\mathbf{Q}^T$ completes the transformation:

$$\mathbf{Q}^T\mathbf{AQ} = \left[\begin{array}{c|c} \mathbf{A}_{n-m,n-m} & (\mathbf{AU})_{n-m,m} \\ \hline (\mathbf{AU})^T_{m,n-m} & \Lambda_{m,m} \end{array}\right] \tag{63a}$$

$$\mathbf{Q}^T\mathbf{BQ} = \left[\begin{array}{c|c} \mathbf{B}_{n-m,n-m} & (\mathbf{BU})_{n-m,m} \\ \hline (\mathbf{BU})^T_{m,n-m} & \mathbf{I}_{m,m} \end{array}\right] \tag{63b}$$

The first subscript $(n-m)$ in the notation $\mathbf{A}_{n-m,n-m}$ and $(\mathbf{AU})_{n-m,m}$ is a reminder that premultiplication of a matrix by $\mathbf{Q}^T$ does not alter the first $(n-m)$ rows of the matrix. The $m \times m$ identity matrix $\mathbf{I}_{m,m}$ is based on the assumption that the eigenvectors have been normalized so that

$$\mathbf{U}_k^T\mathbf{BU}_k = 1 \tag{64}$$

For the normalized eigenvectors, the elements of the diagonal matrix $\Lambda_{m,m}$ are the Rayleigh quotients

$$\lambda_k = \frac{\mathbf{U}_k^T\mathbf{AU}_k}{\mathbf{U}_k^T\mathbf{BU}_k} \tag{65}$$

The presence of the Rayleigh quotients in the transformed equations suggests that the m approximate eigenvalues can be improved by an equivalence tranformation. The orthogonality relation (eq. (58)) can be used to improve the m approximate eigenvectors with a Gram-Schmidt procedure. The subscripts on the m approximate solutions are determined in a definite order by the pivoting strategy in the forward sweep procedure; the ordered subscripts provide a definite order for the Gram-Schmidt procedure.

The ordered subscripts allow the matrix **Q** to be formed column by column, and at the same time the matrices (**QA**), (**QB**), ($\mathbf{Q}^T\mathbf{AQ}$), and ($\mathbf{Q}^T\mathbf{BQ}$) are formed. The formation of **Q** can be summarized as follows:

1. The m column vectors $\mathbf{V}'$ are stored in columns $j = \mathbf{P}_{n+1-k}$ (for $k = 1, 2, ..., m$), where **P** is the permutation vector determined by the pivoting strategy in the forward sweep.
2. The $(n-m)$ column vectors of **Q** have zeroes as elements, except they have ones on the diagonal.
3. The m vectors $\mathbf{V}'$ are defined recursively by a Gram-Schmidt orthogonalization procedure. The procedure is

$$\mathbf{V}_1' = \mathbf{V}_1 \tag{66a}$$

$$\mathbf{V}_2' = \mathbf{V}_2 + g_{12}\mathbf{V}_1' \tag{66b}$$

$$\mathbf{V}_m' = \mathbf{V}_m + \sum_{k=1}^{m-1} g_{km}\mathbf{V}_k' \tag{66c}$$

The coefficients g_{kj} are computed from

$$g_{kj} = \frac{-(\mathbf{V}_k')^T(\mathbf{BV}_j)}{(\mathbf{V}_k')^T(\mathbf{BV}_k')} \tag{67}$$

The denominator in equations (67) is not unity in general, so that

$$(\mathbf{V}_k')^T(\mathbf{BV}_j') = d_{kk}\delta_{kj} \tag{68}$$

The $\mathbf{V}'$ are orthogonal to the $(\mathbf{BV}_j')$ but are not normalized.

The approximate eigenvectors λ are updated with the Rayleigh quotients for $\mathbf{V}_k'$ as follows:

$$\lambda_k' = \frac{(\mathbf{V}_k')^T(\mathbf{AV}_k')}{(\mathbf{V}_k')^T(\mathbf{BV}_k')} \tag{69}$$

In the computer code the primed quantitites λ_k' and $\mathbf{V}_k'$ are stored over the corresponding unprimed variables. The numerator and denominator of equations (69) appear as diagonal elements of $\mathbf{Q}^T\mathbf{AQ}$ and $\mathbf{Q}^T\mathbf{BQ}$. The subscript of the diagonal element is determined from the element $\mathbf{P}_{n+1-k}$ of the permutation vector.

The equivalence transformation for the second numerical example (eqs. (39)) starts with the eigenvectors $\mathbf{V}_1$ and $\mathbf{V}_2$ from equations (49) and (50). The permutation vector $\mathbf{P}$ for the problem is listed in equation (53). For this problem, $n = 4$ and $m = 2$; therefore, $\mathbf{V}_1'$ goes in column $\mathbf{P}(n) = 3$ and $\mathbf{Q}$ and $\mathbf{V}_2'$ goes in column $\mathbf{P}(3) = 2$. The change in variables is

$$\begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \\ \mathbf{X}_4 \end{bmatrix} = \begin{bmatrix} 1 & -0.8369 & -0.4330 & 0 \\ 0 & 0.4960 & 1.000 & 0 \\ 0 & 1.0432 & -0.8229 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \mathbf{Y}_3 \\ \mathbf{Y}_4 \end{bmatrix} \tag{70}$$

The constant g_{12} from equations (67) for the second example is

$$g_{12} = \frac{-0.09796}{1.8646} = -0.0525 \tag{71}$$

The transformed problem $[\mathbf{Q}^T(\mathbf{A} - \lambda\mathbf{B})\mathbf{Q}]\mathbf{Y} = 0$ is

$$\begin{bmatrix} (3-\lambda) & (-0.4752 + 0.8369\lambda) & (-0.1220 + 0.4331\lambda) & 0 \\ (-0.4752 + 0.8369\lambda) & (1.30987 - 2.035\lambda) & (-0.0106 + 0\lambda) & 0 \\ (-0.1220 + 0.4331\lambda) & (-0.0106 + 0\lambda) & (0.5745 - 1.8647\lambda) & 0 \\ 0 & 0 & 0 & (4-\lambda) \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \mathbf{Y}_3 \\ \mathbf{Y}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{72}$$

The new approximations for the lowest $m = 2$ eigenvalues from the Rayleigh quotients (eqs. (69)) are

$$\lambda_1' = \frac{(\mathbf{V}_1')^T(\mathbf{AV}_1')}{(\mathbf{V}_1')^T(\mathbf{BV}_1')} = \frac{0.5745}{1.8647} = 0.308079 \tag{73a}$$

$$\lambda_2' = \frac{(\mathbf{V}_2')^T(\mathbf{AV}_2'^{2})}{(\mathbf{V}_2')^T(\mathbf{BV}_2'^{2})} = \frac{1.30987}{2.035} = 0.64376 \tag{73b}$$

The first eigenvalue λ_1' is a very good approximation of the lowest eigenvalue for equations (39). An approximate check on λ_1' is possible since the exact value of $\lambda = \lambda_1$ makes the determinant of the coefficients of equations (72) vanish. Substitution of $\lambda = \lambda_1'$ from equations (73a) into (72) makes the coefficients of the third row (or column) zero or near zero. An identically zero row (or column) is a sufficient condition for a determinant to be zero.

The approximate check on the eigenvalue is the basis for the final step in the first iteration cycle for the second numerical example. The final step is contained in the next section of this report.

The numerical example has $n = 4$ and $m = 2$, but the equivalence transformation for the example illustrates the matrix algebra for the general case. For the general case of m approximate eigenvectors of an n-dimensional problem, the matrix of the equivalent problem $[\mathbf{Q}^T(\mathbf{A} - \lambda\mathbf{B})\mathbf{Q}]\mathbf{Y} = 0$ differs from the original problem by only m rows and m columns. The m rows and columns are determined by use of a permutation vector defined in the forward sweep procedure. In each of the m columns that are changed, $(n-m)$ elements are simply the elements of $(\mathbf{A} - \lambda\mathbf{B})$ times the corresponding approximate eigenvector. These elements are also stored for use as part of the Gram-Schmidt orthogonalization procedure. The coefficients in an $m \times m$ submatrix of the transformed problem are also found in the orthogonalization procedure. Therefore, starting with the eigenvector with the lowest subscript, the rules of matrix multiplication allow efficient computation and storage of the equivalence transformation and of the coefficients in the orthogonalization procedure simultaneously.

Modified Sweep Procedures

The modified sweep procedures are the last step in the iteration cycle for delayed division. The equivalence transformation and the modified sweep procedures form an inner iteration loop for improving the m approximate eigenvalues computed in the forward sweep and the m approximate eigenvectors computed in the back substitution. (See fig. 1.)

After the equivalence transformation, the modified sweep procedures operate on the transformed problem

$$(\mathbf{A}' - \lambda\mathbf{B}')\mathbf{Y} = 0 \tag{74}$$

where

$$\mathbf{A}' = \mathbf{Q}^T\mathbf{A}\mathbf{Q}$$

$$\mathbf{B}' = \mathbf{Q}^T\mathbf{B}\mathbf{Q}$$

At this stage of the iteration, considerable information is available about the eigenvalue problem. The purpose of the modified sweep procedures is to use the information to reduce the number of computer operations in the iterative solution. Some information is related to exact properties of the problem and some information is related to only approximate properties. One exact property is that the transformed problem has the same eigenvalues as the original problem $(\mathbf{A} - \lambda\mathbf{B})\mathbf{X} = 0$, whereas the eigenvectors are related through the change in variables $\mathbf{X} = \mathbf{Q}\mathbf{Y}$. Because of the form of $\mathbf{Q}$, only the elements of m rows and m columns of matrix $\mathbf{A}'$ differ from the corresponding elements of matrix $\mathbf{A}$, with the remaining $(n-m)^2$ elements identical for $\mathbf{A}'$ and $\mathbf{A}$. A similar statement holds for $\mathbf{B}'$ and $\mathbf{B}$.

Pertinent information on approximate properties is that the matrices $\mathbf{A}''$ defined by

$$\mathbf{A}'' = \mathbf{A}' - \lambda_k\mathbf{B}' \qquad (k = 1, 2, ..., m) \tag{75}$$

contain at least one row and column that are nearly zero. The Gram-Schmidt orthogonalization procedure used in forming the matrix $\mathbf{Q}$ can be viewed as minimizing one column of each of the $\mathbf{A}''$ matrices in the least-squares sense. The nonzero elements in each of the m columns are a measure of the residual error in the orthogonalization.

The matrices $\mathbf{A}''$ in equations (75) suggest that a shift in origin for each of the m eigenvalues would speed convergence of the iteration for solutions of equations (74). The modified sweep procedure makes the eigenvalue shifts simultaneously rather than sequentially. The simultaneous shift takes advantage of the fact that $(n-m)^2$ elements of $\mathbf{A}$ and of $\mathbf{B}$ are unchanged by the transformation to $\mathbf{A}'$ and to $\mathbf{B}'$.

The operations for taking advantage of the unchanged elements are stored during the forward sweep procedure on the original problem $(\mathbf{A} - \lambda\mathbf{B})\mathbf{X} = 0$. In matrix notation, the forward sweep procedure results in

$$(\mathbf{L}^{-1} + \lambda\mathbf{M})(\mathbf{A} - \lambda\mathbf{B}) = (\mathbf{R} + \lambda\mathbf{S}) + O(\lambda^2) \tag{76}$$

The notation in equations (76) is an extension of the notation for Gaussian elimination subroutines. The matrix **R** is upper triangular and **L** is lower triangular. They are factors of **A**:

$$\mathbf{A} = \mathbf{LR} \tag{77}$$

The matrix **R** is stored in the upper right-hand half of auxiliary matrix **C** during the forward sweep, whereas $\mathbf{L}^{-1}$ is stored in the lower triangular part of **C** as a product of elementary matrices. The matrices **M** and **S** in equations (76) are extensions of Gaussian elimination. Matrix **S** is upper triangular and is stored explicitly during the forward sweep, whereas **M** is lower triangular. Matrix **M** is stored as a product of elementary matrices derived by dropping quadratic terms in λ at each elimination step in the forward sweep.

After the equivalence transformation, it can be shown that $(n-m)$ rows and columns of matrices $(\mathbf{L}^{-1}+\lambda\mathbf{M})$ and $(\mathbf{R}+\lambda\mathbf{S})$ are unchanged. Therefore, for small λ, any eigenvalue shift can be applied to submatrices of $(\mathbf{L}^{-1}+\lambda\mathbf{M})$ and $(\mathbf{R}+\lambda\mathbf{S})$. If the shift is defined by

$$\lambda = \lambda' + \mu \tag{78}$$

the approximation is

$$(\mathbf{L}^{-1} + \lambda\mathbf{M}) = (\mathbf{L}^{-1} + \lambda'\mathbf{M}) + \mu\mathbf{M} \tag{79a}$$

$$(\mathbf{R} + \lambda'\mathbf{S}) = (\mathbf{R} + \lambda'\mathbf{S}) + \mu\mathbf{S} \tag{79b}$$

The approximation in equations (79) is used in the modified sweep procedures for eliminating the first $(n-m)$ variables in the transformed problem (eqs. (74)). Using the approximation in equations (79) avoids repeating numerical operations in the sweep for the transformed problem since the results of the operations have been stored during the forward sweep on the original problem.

In the m columns of the matrices in the transformed problem $(\mathbf{A}' - \lambda\mathbf{B}')\mathbf{Y} = 0$ that are different from the original problem, the simultaneous shift is written explicitly as

$$\lambda = \lambda_k + \mu_k \qquad (k = 1, 2, ..., m) \tag{80}$$

where λ_k are current approximate eigenvalues stored after the equivalence transformation. The values of μ_k remain to be determined in the modified sweep procedure. In operating on any one of the m columns with elementary matrices of the form of equations (79a), the corresponding λ_k value is substituted for λ'. Because of the way elements are stored for the modified back substitution, a convenient identity (derived from eq. (80)) once a value is assigned to one of the $\mu_k = \mu$ is

$$\mu_i = (\lambda_k - \lambda_i) + \mu_k \tag{81}$$

The modified procedures can be thought of as partitioning the transformed problem. In an $(n-m)\times(n-m)$ submatrix of the transformed problem, convergence is quadratic in λ_k. In the rest of the transformed problem, the convergence is faster since it is quadratic in μ_k.

The second numerical example (eqs. (39)) has $n = 4$ and $m = 2$. The value of n is too small to be a good illustration of equations (79), but the application of the simultaneous shift in equation (81) is clear. The transformed problem $(\mathbf{A}' - \lambda\mathbf{B}')\mathbf{Y} = 0$ is written out in equations (72) after the change in variables $\mathbf{X} = \mathbf{QY}$ in equations (70). The eigenvalue shift in equation (80) is

$$\lambda = \lambda_1 + \mu_1 = 0.308079 + \mu_1 \tag{82a}$$

$$\lambda = \lambda_2 + \mu_2 = 0.64376 + \mu_2 \tag{82b}$$

The $m = 2$ columns of the transformed problem that are shifted are columns 3 and 2. Equations (72) are rewritten

$$\begin{bmatrix} (3-\lambda) & (0.06357 + 0.8369\mu_2) & (0.01143 + 0.4331\mu_1) & 0 \\ (-0.4752 + 0.8369\lambda) & (0 - 2.035\mu_2) & (-0.0106 + 0\mu_1) & 0 \\ (-0.1220 + 0.4331\lambda) & (-0.0106 + 0\mu_2) & (0 - 1.8647\mu_1) & 0 \\ 0 & 0 & 0 & (4-\lambda) \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \mathbf{Y}_3 \\ \mathbf{Y}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{83}$$

The elimination of $\mathbf{Y}_1$ and $\mathbf{Y}_4$ in equations (83) is the same as the elimination of $\mathbf{X}_1$ and $\mathbf{X}_4$ for the original problem (eqs. (39)). The forward sweep stores elements necessary to generate the elementary matrix (see eqs. (40))

$$\left(\mathbf{C}_{14}^{-1} + \lambda \mathbf{D}_{14}^{-1}\right) = \begin{bmatrix} (1/3) + (\lambda/9) & 0 & 0 & (0 + 0\lambda) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ (0 + 0\lambda) & 0 & 0 & (1/4) + (\lambda/16) \end{bmatrix} \tag{84}$$

The transformed equations are premultiplied by $(\mathbf{C}_{14}^{-1} + \lambda \mathbf{D}_{14}^{-1})$. In forming elements of columns 1 and 4 of the product no shift in $(\mathbf{C}_{14}^{-1} + \lambda \mathbf{D}_{14}^{-1})$ is used, whereas column 3 elements use the shift in equation (82a) and column 2 elements are computed from the shift in equation (82b). Which column to use in the shift with λ is deduced from the permutation vector (eq. (53)). The result of the premultiplication of equations (83) by $(\mathbf{C}_{14}^{-1} + \lambda \mathbf{D}_{14}^{-1})$, after dropping quadratic terms in λ, μ_1, and μ_2, is

$$\begin{bmatrix} 1 & (0.0257 + 0.3459\mu_2) & (0.0042 + 0.1604\mu_1) & 0 \\ (-0.4752 + 0.8369\lambda) & (0 - 2.035\mu_2) & (-0.0106 + 0\mu_1) & 0 \\ (-0.1220 + 0.4331\lambda) & (-0.0106 + 0\mu_2) & (0 - 1.8647\mu_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \mathbf{Y}_3 \\ \mathbf{Y}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{85}$$

The elimination of $\mathbf{Y}_1$ and $\mathbf{Y}_4$ from equation (85) is completed with premultiplication by the elementary matrix $(\mathbf{L}_{14}^{-1} + \lambda \mathbf{M}_{14})$:

$$\left(\mathbf{L}_{14}^{-1} + \lambda \mathbf{M}_{14}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -(-0.4752 + 0.8369\lambda) & 1 & 0 & -(0 + 0\lambda) \\ -(-0.1220 + 0.4331\lambda) & 0 & 1 & -(0 + 0\lambda) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{86}$$

The premultiplication by $\left(\mathbf{L}_{14}^{-1} + \lambda \mathbf{M}_{14}\right)$ follows the same pattern as the premultiplication by $\left(\mathbf{C}_{14}^{-1} + \lambda \mathbf{D}_{14}^{-1}\right)$ for eigenvalue shifts and truncation of quadratic terms. The result of the operations transforms equations (85) into

$$\begin{bmatrix} 1 & (0.0257 + 0.3459\mu_2) & (0.0042 + 0.1604\mu_1) & 0 \\ 0 & (-0.0016 - 2.078\mu_2) & (-0.0097 + 0.0314\mu_1) & 0 \\ 0 & (-0.0146 - 0.0653\mu_2) & (-0.000048 - 1.868\mu_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \mathbf{Y}_3 \\ \mathbf{Y}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{87}$$

The modified sweep procedure ends at this stage after eliminating $(n - m) = 2$ variables. A total of $m = 2$ equations are still coupled, but the coupling is weak. A relaxation type of algorithm is followed to compute the corrections μ_1 and μ_2 to the eigenvalues. Although the modified sweep has not produced a complete upper triangular matrix, the off-diagonal terms are small enough to neglect. Neglecting off-diagonal terms in the lower triangular matrix has the effect of reversing the order of the Gram-Schmidt orthogonalization procedure in the equivalence transformation. The orthogonalization procedure sweeps lower numbered approximate eigenvectors from higher numbered eigenvectors. The relaxation procedure sweeps higher numbered eigenvectors from lower modes. In the general case, the triangular equation form is in terms of equations reordered by the permutation vector. Equations (87) happen to be triangular without reordering the equations when the element in the third row and second column is neglected. The determinant of the triangular form vanishes when the diagonal elements are zero. The corrections to the eigenvalues have the following values that make the diagonal elements vanish:

$$\mu_1 = -\frac{(-0.000047595)}{(-1.868297)} = -2.5475 \times 10^{-5} \tag{88a}$$

$$\mu_2 = -\frac{(-0.0016328)}{(-2.078)} = -7.857083 \times 10^{-4} \tag{88b}$$

After the corrections to the eigenvalues, μ_k, are determined, the eigenvectors for the transformed problem are computed with the modified back substitution procedure. This modified procedure is similar to the original back substitution procedure except that equation (81) must be used to relate the eigenvalue corrections that appear in m columns.

The completion of the modified back substitution is the end of the first full iteration cycle for the second numerical example. The equivalence transformation and modified sweep procedures form an inner iteration cycle that can be repeated with stored data from the forward sweep procedure.

The results for the second numerical example with one forward sweep and two modified sweeps are summarized in tables I and II.

TABLE I. TWO LOWEST EIGENVALUES OF EQUATIONS (39)

Operation	λ_1	λ_2
First forward sweep	0.31101776	0.68898224
First Rayleigh quotient	.30807949	.64376187
First modified sweep	.30805401	.64297616
Second Rayleigh quotient	.30797841	.64310402
Second modified sweep	.30797867	.64310399
Exact solution	0.30797853	0.64310413

TABLE II. TWO LOWEST EIGENVECTORS OF EQUATIONS (39)

Component of eigenvector	Value from—			Exact solution
	First back substitution	First modified back substitution	Second modified back substitution	
u_{11}	−0.433053	−0.445546	−0.44502	−0.445042
u_{12}	1.0	1.0	1.0	1.0
u_{13}	−.822876	−.802830	−.801976	−.801938
u_{14}	0	0	0	0
u_{21}	−0.859602	−0.826558	−0.802697	−0.801938
u_{22}	.548584	.475492	.445928	.445042
u_{23}	1.0	1.0	1.0	1.0
u_{24}	0	0	0	0

The convergence of the method of delayed division is rapid. As might be expected from dropping quadratic terms in λ at each step, the lowest eigenvalue and lowest mode shape converge faster than the second solution.

Summary of Delayed Division

Each step in the method of delayed division has been outlined separately in the previous sections of this report. In the actual computations the steps are independent, but certain data are stored for use in later steps. This section briefly reviews the data storage in the SPLIT subroutine as the storage relates to avoiding duplicate calculations.

The storage requirements in the forward sweep subroutine parallel the storage in Gaussian elimination subroutines for the linear algebra problem $\mathbf{AX} = f$. The main difference in storage requirements is that the forward sweep procedure operates on two matrices $\mathbf{A}$ and $\mathbf{B}$ in the eigenproblem $(\mathbf{A} - \lambda\mathbf{B})\mathbf{X} = 0$. The matrices $\mathbf{A}$ and $\mathbf{B}$ are stored for later use in the equivalence transformation. The forward sweep procedure operates on two auxiliary matrices $\mathbf{C}$ and $\mathbf{D}$. At the start of the forward sweep procedure, $\mathbf{A} = \mathbf{C}$ and $\mathbf{D} = -\mathbf{B}$. As presented in equation (76), the forward sweep subroutine operates on matrices $\mathbf{C}$ and $\mathbf{D}$ to produce the form

$$(\mathbf{L}^{-1} + \lambda\mathbf{M})(\mathbf{A} - \lambda\mathbf{B}) = \mathbf{R} + \lambda\mathbf{S} + O(\lambda^2)$$

At the end of the forward sweep routine, elements of the upper triangular matrix $\mathbf{R}$ are stored in matrix $\mathbf{C}$. Because of the pivoting strategy in the forward sweep, matrix $\mathbf{R}$ is not stored in the upper right-hand triangle of matrix $\mathbf{C}$. A separate permutation vector is stored that can formally generate elementary permutation matrices that can form $\mathbf{R}$ as a matrix product. However, the permutation vector locates the elements of $\mathbf{R}$ so that $\mathbf{R}$ is never arranged explicitly in triangular form. The elements of the elementary matrices whose product is $\mathbf{L}^{-1}$ are also stored in $\mathbf{C}$. Elements that are known to be one or zero are not stored. There is also room in matrix $\mathbf{C}$ to store factors in the determinant of $\mathbf{A}$. Also, in the forward sweep, the elements of the triangular matrix $\mathbf{S}$ are stored in matrix $\mathbf{D}$ along with elements of elementary matrices necessary to form the lower triangular matrix $\mathbf{M}$.

The forward sweep is followed by the back substitution subroutine. The back substitution computes approximate eigenvectors $\mathbf{V}_k$ by explicitly solving

$$(\mathbf{R} + \lambda_k \mathbf{S})\mathbf{V}_k = 0 \tag{90}$$

The eigenvectors are stored in a matrix $\mathbf{U}$ with m columns and n rows.

The back substitution subroutine is followed by the equivalence transformation. The transformed problem is $\mathbf{Q}^T(\mathbf{A} - \lambda\mathbf{B})\mathbf{Q}\mathbf{Y} = 0$. The matrix $\mathbf{Q}$ and its transpose $\mathbf{Q}^T$ are not stored since the elements of $\mathbf{Q}$ are ones and zeroes and the m columns are stored in the matrix $\mathbf{U}$. Most of the elements of the product $\mathbf{Q}^T\mathbf{A}\mathbf{Q}$ are identical with matrix $\mathbf{A}$. The m rows and columns that are different are stored in matrix $\mathbf{C}$ and similar terms from the product $-\mathbf{Q}^T\mathbf{B}\mathbf{Q}$ are stored in matrix $\mathbf{D}$. The m rows and columns are determined by the last m elements of the permutation vector.

The choice of storage locations partitions the transformed problem $\mathbf{Q}^T(\mathbf{A} - \lambda\mathbf{B})\mathbf{Q}\mathbf{Y} = 0$. A forward sweep eliminating the first $(n - m)$ variables of the transformed problem would duplicate the same computations in the original forward sweep. However, the modified sweep procedure goes directly to the $(n - m)$ rows and columns of matrices $\mathbf{C}$ and $\mathbf{D}$ and uses the $(\mathbf{L}^{-1} + \lambda\mathbf{S})$ data that were stored on the forward sweep. Only m modal amplitudes remain to be eliminated on the modified sweep. The partitioning is such that quadratic terms in λ_k are dropped in $(n - m)$ columns of the modified forward sweep whereas quadratic terms in μ_k are dropped in m columns, where μ_k is the correction on the current approximation for λ_k.

The modified back substitution computes approximate eigenvectors $\mathbf{W}_k$ for the transformed problem. Only one vector $\mathbf{W}$ is stored at one time and is used to write the vector $\mathbf{Q}\mathbf{W}_k$. The vector $\mathbf{Q}\mathbf{W}_k$ is stored in matrix $\mathbf{U}$, updating the jth eigenvector $\mathbf{V}_k$ for the original problem. Not storing all the transformed eigenvectors $\mathbf{W}_k$ modifies $\mathbf{Q}$ so that the new approximate eigenvectors $\mathbf{V}_k$ are not computed exactly from the eigenvectors of the transformed problem. However, the approximation for the eigenvectors is consistent with the Gram-Schmidt orthogonalization procedure used to form the matrix $\mathbf{U}$ at the end of the forward sweep. The Gram-Schmidt procedure sweeps components of lower numbered approximate eigenvectors from higher numbered eigenvectors. The modified back substitution reverses the order and sweeps higher numbered eigenvectors from lower numbered approximate eigenvectors.

The subroutines contained in the SPLIT subroutine make a complete iterative procedure. Convergence is problem dependent and is based on the assumption that m eigenvalues are small in absolute value compared with the remaining eigenvalues. In general, small eigenvalues converge faster than larger eigenvalues. The eigenvalues converge faster than the eigenvectors. The iteration is unaffected by multiple eigenvalues or closely spaced eigenvalues.

These general conclusions are based on the theory and on results for some example problems presented in the next section of this report. After the example problems, the method of delayed division is compared with other methods for solving the symmetric eigenvector problem $(\mathbf{A} - \lambda\mathbf{B})\mathbf{X} = 0$.

Example Problems

This section contains four eigenproblems solved with the method of delayed division. The problems are selected to illustrate different features of the method. The first problem is a column buckling problem modeled by finite differences. The column problem is a typical applied problem in which the matrix $\mathbf{B}$ of the problem $(\mathbf{A} - \lambda\mathbf{B})\mathbf{X} = 0$ is not an identity matrix.

The second example problem is a tridiagonal matrix problem posed by Wilkinson (ref. 4). Wilkinson uses the problem to show sources of numerical round-off errors in computing eigenvectors. However, the pivoting strategy in the method of delayed division is numerically stable and the computed eigenvectors are correct for the example problem.

The third example problem requires more analysis. The problem is computing shear buckling loads for plates. The transverse plate deflection for a simply supported plate is represented in the numerical solution by a double Fourier sine series. The eigenvalue is proportional to the shear buckling load. The physics of the problem dictates that for every positive eigenvalue there is a negative eigenvalue of the same absolute value. The occurrence of pairs of eigenvalues means that the strategy of dropping quadratic terms in delayed division fails for the shear problem. Two alternate methods for making linear terms dominant in the shear buckling problem are shown.

The fourth example problem illustrates how an understanding of the method of delayed division allows the algorithm to be adapted to more sophisticated problems. The fourth problem applies the Lagrangian multiplier method to plate buckling. Through the use of Lagrangian multipliers, the order of the matrices **A** and **B** in the problem $(\mathbf{A} - \lambda\mathbf{B})\mathbf{X} = 0$ is greatly reduced with no loss of accuracy in the final results.

Column Buckling Problem

The differential equation from linear theory for buckling of a column is

$$EI\,\frac{d^4y}{dx^4} + P\,\frac{d^2y}{dx^2} = 0 \tag{91}$$

(See, for example, ref. 5.) The example problem is to compute the axial load P for a nontrivial solution of the differential equation subject to the boundary conditions for a clamped column. These boundary conditions at $x = 0$ and $x = L$ are

$$y = 0$$

$$\frac{dy}{dx} = 0$$

The numerical eigenvalue problem is formulated by approximating the differential equation by central differences at N interior nodes:

$$y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2} + \lambda(y_{i-1} - 2y_i + y_{i+1}) = 0 \qquad (i = 1, 2, ..., N) \tag{92}$$

The difference equation is written for N interior points. The eigenvalue λ is

$$\lambda = \frac{PL^2}{EI}\frac{1}{(N+1)^2} = \frac{P\pi^2}{P_E(N+1)^2} \tag{93}$$

where $P_E = \pi^2 EI/L^2$ is the Euler load for a simply supported column. The difference approximations for the boundary conditions are

$$\left.\begin{array}{lll} y_0 = 0 & \text{and} & y_{N+1} = 0 \\ y_{-1} = y_1 & \text{and} & y_{N+2} = y_N \end{array}\right\} \tag{94}$$

The unknowns on the left-hand sides of equations (94) are eliminated from equations (92), with N homogeneous equations left in N unknown y_i at the interior nodes.

The numerical results from the SPLIT subroutine are summarized in table III. The effect on the critical loads P_k produced by varying N is summarized in table III. As N increases, the critical loads from the numerical eigenvalue solution approach the buckling loads from the exact solution of the differntial equation (91). The odd-numbered critical loads for fixed N can be checked from the exact formula

$$\frac{P_k}{P_E} = 2\left(1 - \cos\frac{k\pi + \pi}{N+1}\right)\left(\frac{N+1}{\pi}\right)^2 \tag{95}$$

Equations (95) follow from an exact solution of the linear difference equations (92). The error tolerance on convergence for the iteration in the SPLIT subroutine was set to give six-digit accuracy for the critical loads. The lower buckling loads listed in table III hold this accuracy, and up to six eigenvalues were computed with good accuracy for each forward sweep.

TABLE III. CRITICAL LOADS AS A FUNCTION OF N

(a) Odd-numbered loads

N	$(P_1/P_E)^*$	$(P_1/P_E)^\dagger$	$(P_3/P_E)^*$	$(P_3/P_E)^\dagger$	$(P_5/P_E)^*$	$(P_5/P_E)^\dagger$
4	3.50056	3.50056				
9	3.87012	3.87012				
10	3.89242	3.89242	14.3338	14.3339		
20	3.97025	3.97025	15.5282	15.5282	33.6473	33.6469
50	3.99494	3.99494	15.9192	15.9192	35.5928	35.5921
100	3.99871	3.99871	15.9794	15.9794	35.8956	35.8956
∞		4.00000		16.0000		36.0000

*From SPLIT subroutine.
†From equations (95).

(b) Even-numbered loads (from SPLIT subroutine)

N	(P_2/P_E)	(P_4/P_E)	(P_6/P_E)
4	5.874		
9	7.544		
10	7.652		
20	8.0344	23.0447	43.89
50	8.1577	23.9985	47.44
100	8.1765	24.1369	47.99
Exact	8.1830	24.2872	48.19

However, the example problem did reveal a limitation of the Gram-Schmidt orthogonalization procedure. The orthogonalization procedure is used to sweep lower modes out of approximate eigenvectors. The eigenvectors are columns in matrix $\mathbf{Q}$ in the equivalence transformation $\mathbf{Q}^T(\mathbf{A} - \lambda\mathbf{B})\mathbf{Q}$. To compute 10 or 20 eigenvectors in the matrix $\mathbf{Q}$ for the column buckling problem, the orthogonalization exhibited numerical instability. Although the eigenvalues are all distinct for the column buckling problem, the orthogonalization procedure failed to retain linear independence in the approximate eigenvectors. If linear independence is not maintained, the inner iteration loop fails to converge.

The limitation imposed by the orthogonalization procedure on the number of eigenvalues and eigenvectors that can be computed with one forward sweep is not serious for postbuckling problems and for many other problems. In postbuckling problems, only a small number of nearly equal eigenvalues and eigenvectors are required for the analysis at any given load level. Therefore, the SPLIT subroutine is efficient for postbuckling problems. For problems in which more than a few eigenvectors are required, the orthogonalization procedure in SPLIT should be modified. The modification to the equivalence transformation is discussed in the section entitled *Delayed Division and Other Methods.*

The efficiency of the iteration in SPLIT is indicated by computer times listed in table IV for executing different subroutines in the SPLIT subroutine. The times are CPU time in seconds on the CDC CYBER 175 computer at the NASA Langley Research Center; however, the relative amount of time spent in each subroutine should be relevant for other computers. The forward sweep subroutine is executed once. It is the main subroutine in the method of delayed division. The forward sweep produces the factored form in equations (76) and includes the pivoting strategy and permutation vector described in the section entitled *Forward Sweep.* As might be expected from analogous results for Gaussian elimination subroutines on matrices of order n, the computer time for the forward sweep is roughly proportional to n^3.

TABLE IV. APPROXIMATE COMPUTER TIME FOR SUBROUTINES ON CDC CYBER 175

			Subroutine CPU time for each call, sec, for—		
n[*]	m[†]	M[‡]	Forward sweep	Equivalence transformation	Modified sweep
4	2	6	0.020	0.0075	0.010
9	2	7	.025	.0150	.020
10	2	5	.040	.0150	.020
10	4	9	.040	.0200	.025
20	2	5	.170	.0250	.060
20	4	11	.175	.0450	.110
20	6	14	.175	.0700	.150
50	2	3	2.255	.1000	.325
50	4	5	2.275	.2100	.645
50	10	16	2.285	.5450	1.525
100	2	2	17.451	.3650	1.280
100	10	16	17.856	2.0400	6.255
100	20	16	18.095	4.3500	12.300

[*] Order of matrices **A** and **B**.

[†] Number of computed eigenvalues and eigenvectors.

[‡] Number of iterations calling modified sweep subroutine and subroutine for orthogonalization and Rayleigh quotient.

The equivalence transformation subroutine and the modified sweep subroutine are iterative and are each called by SPLIT a total of M times. The execution time for these iterative subroutines is approximately proportional to the product mn^2, where m is the number of approximate eigenvectors in the equivalence transformation.

The main conclusion that can be drawn from table IV is that the forward sweep procedure is efficient compared with the other subroutines. For matrices of the order of $n = 100$ or less, using $m = 6$ and then shifting to compute higher eigenvalues ($m > 6$) is an efficient strategy for SPLIT.

Eigenvector Stability

The column buckling problem revealed a numerical instability in the equivalence transformation if m, the number of computed eigenvectors per forward sweep, is too large. Wilkinson (ref. 4) discusses another type of numerical instability that can occur in computations for eigenvalue problems. This instability is manifested by errors in computed eigenvectors from known eigenvalues. Wilkinson recommends using a pivoting strategy for computation of eigenvectors. The complete pivoting in the forward sweep subroutine is numerically stable so that the eigenvectors computed by the SPLIT subroutine are numerically stable whenever the iteration for the associated eigenvalue converges.

The example problem used here to illustrate the stable eigenvector computation is taken from reference 4. The problem has the simple tridiagonal form

$$(\alpha_1 - \lambda)x_1 + \beta_1 x_2 = 0 \tag{96a}$$

$$\beta_i x_{i-1} + (\alpha_i - \lambda)x_i + \beta_{i+1}x_{i+1} = 0 \qquad (i = 2, 3, 4, ..., (n-1)) \tag{96b}$$

$$\beta_n x_{n-1} + (\alpha_n - \lambda)x_n = 0 \tag{96c}$$

The particular tridiagonal form used in this example is

$$\alpha_i = -i + 1 \quad (i = 1, 2, ..., n) \tag{97a}$$

$$\beta_i = 1 \tag{97b}$$

The terms α_i in equations (97a) are shifted by 10 from Wilkinson's problems so that the smallest λ_1 of 0.74619418 for equations (96) with $n = 21$ corresponds to the largest λ_1 of 10.74619418 of reference 4. The exact eigenvectors are not affected by the shift.

Computed eigenvectors associated with λ_1 are listed in table V. The first column and the third column are almost identical. The first column is the first eigenvector from the SPLIT subroutine operating on the shifted problem $[(\mathbf{A} - \lambda_1\mathbf{I}) - \lambda\mathbf{I}]\mathbf{X} = 0$, whereas the third column is reported by Wilkinson using pivoting for size on the matrix $(\mathbf{A} - \lambda_1\mathbf{I})$. The second column is from the SPLIT subroutine operating on the problem $(\mathbf{A} - \lambda_1\mathbf{I})\mathbf{X} = 0$. The second column is not quite as accurate as the first column, but the computed eigenvector is numerically stable. The fourth column is the eigenvector computed by Wilkinson directly from the matrix $(\mathbf{A} - \lambda_1\mathbf{I})$ without pivoting. The top components are correct, but round-off error makes the bottom components incorrect.

TABLE V. STABILITY OF EIGENVECTOR COMPUTATION DUE TO PIVOTING IN DELAYED DIVISION

[$\lambda_1 = 0.74219418$, first eigenvector u from delayed division]

i	u_i for—			
	$\bar{\lambda} = \lambda_1$	$\bar{\lambda} = 0$	Pivoting*	No pivoting*
1	1.0000000×10^{0}	1.0000000×10^{0}	1.00000000	1.00000000
2	7.4619418×10^{-1}	7.4627843×10^{-1}	.74619418	.74619420
3	3.0299994×10^{-1}	3.0284026×10^{-1}	.30299994	.30299998
4	8.5902494×10^{-2}	8.5305094×10^{-2}	.08590249	.08590260
5	1.8807481×10^{-2}	1.8413750×10^{-2}	.01880748	.01880783
6	3.3614646×10^{-3}	3.2654781×10^{-3}	.00336146	.00336303
7	5.0814709×10^{-4}	4.8526901×10^{-4}	.00050815	.00051681
8	6.6594309×10^{-5}	6.2996079×10^{-5}	.00336146	.00012346
9	7.7053623×10^{-6}	7.1409899×10^{-6}	.00000771	.0004395
10	7.9828544×10^{-7}	7.3152313×10^{-7}	.00000080	.003721
11	7.4882661×10^{-8}	6.6978590×10^{-8}	.00000007	.03582
12	6.4181726×10^{-9}	5.6654776×10^{-9}	.00000001	.3812
13	$5.0644124 \times 10^{-10}$	$4.3461741 \times 10^{-10}$	Neglected	4.442×10^{0}
14	$3.7025740 \times 10^{-11}$	$3.1292977 \times 10^{-11}$		5.624×10^{1}
15	$2.5217695 \times 10^{-12}$	$2.0630252 \times 10^{-12}$		7.686×10^{2}
16	$1.6076278 \times 10^{-13}$	$1.2922596 \times 10^{-13}$		1.128×10^{4}
17	$9.6324699 \times 10^{-15}$	$7.4583540 \times 10^{-15}$		1.1768×10^{5}
18	$5.4443172 \times 10^{-16}$	$4.1313141 \times 10^{-16}$		2.950×10^{6}
19	$2.9121116 \times 10^{-17}$	$2.1168481 \times 10^{-17}$		5.217×10^{7}
20	$1.4783799 \times 10^{-18}$	$1.0500629 \times 10^{-18}$		9.751×10^{8}
21	$7.1260294 \times 10^{-20}$	$4.8196742 \times 10^{-20}$	↓	1.920×10^{10}

*From ref. 4.

The fact that the example is a simple tridiagonal problem does not affect the general conclusion that the eigenvector computation in the method of delayed division is numerically stable. As with other iterative methods, the components of the eigenvectors computed by delayed division are not as accurate to as many digits as are the computed eigenvalues.

In addition to the results already listed, results were computed for equations (96) with different input for m. As in the column buckling problem of the previous section, a value of $m = 6$ gives good numerical results. For $m = 10$, the orthogonalization procedure fails to preserve linear independence in the higher modes.

Shear Buckling in Plates

The third example is an applied problem on the shear buckling in orthotropic, simply supported plates. The matrix equations are generated from a Galerkin solution of the linear buckling equation with a double Fourier sine series used for the transverse deflection. The algebraic equations for the special case for an isotropic plate reduce to those solved by Stein and Neff (ref. 6) by power iteration.

The linear buckling analysis is the first step in nonlinear postbuckling analysis. The equivalence transformation applied in the eigenvalue computation is also useful for the nonlinear problem. However, this report only considers the linear buckling problem.

The forward sweep procedure in delayed division systematically retains linear terms in the eigenvalue while dropping quadratic terms. Delayed division fails when it is applied directly to the shear buckling problem because only quadratic terms appear in the forward sweep. The matrix **D** at the end of the forward sweep is zero. The analysis that follows shows that the terms for the shear load in the shear buckling problem are quadratic. The analytical results that shear buckling loads occur in pairs of opposite sign is not surprising on physical grounds, since a change in sign of a shear load merely indicates a change in direction. The insight provided by the analysis of the form of matrices **A** and **B** for the shear problem and by the fact that linear terms in the eigenvalue are retained in SPLIT allows modification of matrices **A** or **B** or both.

The governing differential equation for the plate buckling problem is

$$\begin{aligned} D_{11}\frac{\partial^4 w}{\partial x^4} + 2(D_{12} + 2D_{66})\frac{\partial^4 w}{\partial x^2 y^2} + D_{22}\frac{\partial^4 w}{\partial y^4} \\ = N_x\frac{\partial^2 w}{\partial x^2} + N_y\frac{\partial^2 w}{\partial y^2} + 2N_{xy}\frac{\partial^2 w}{\partial x\,\partial y} \end{aligned} \tag{98}$$

For the numerical example, all the coefficients in equation (98) are assumed constant. A Galerkin solution is sought in the form of a double Fourier sine series that satisfies the boundary conditions of simple support term by term. Thus,

$$w = \sum_{p=1}^{\infty}\sum_{q=1}^{\infty} w_{pq} \sin\frac{p\pi x}{a} \sin\frac{q\pi y}{b} \tag{99}$$

The partial differential equation (98) is satisfied if the Fourier coefficients w_{pq} are solutions of

$$k_{pq}w_{pq} - \kappa\sum_{r=1}^{\infty}\sum_{s=1}^{\infty}\gamma_{pqrs}w_{rs} = 0 \qquad \left\{\begin{array}{l} p = 1, 2, \ldots, \infty) \\ (q = 1, 2, \ldots, \infty) \end{array}\right. \tag{100}$$

where

$$k_{pq} = \left\{p^4 + 2\left[\left(\frac{D_{12}}{D_{11}}\right) + \left(\frac{2D_{66}}{D_{11}}\right)\right] p^2q^2\beta^2 + q^4\beta^4\left(\frac{D_{22}}{D_{11}}\right)\right\} - K_x p^2\beta^2 - K_y q^2\beta^2$$

$$\beta = \frac{a}{b}$$

$$K_x = \frac{N_x b^2}{\pi^2 D_{11}}$$

$$K_y = \frac{N_y b^2}{\pi^2 D_{11}}$$

The coupling terms γ_{pqrs} are

$$\gamma_{pqrs} = 0 \qquad (p = r \text{ or } \ q = s) \tag{101a}$$

$$\gamma_{pqrs} = \frac{pqrs}{(p^2 - r^2)(q^2 - s^2)} \; \frac{[1 - (-1)^{p+r}][1 - (-1)^{q+s}]}{4} \qquad (p \neq r \text{ or } \ q \neq s) \tag{101b}$$

The eigenvalue to be determined is κ:

$$\kappa = \frac{32\beta K_s}{\pi^2} \tag{102}$$

where $K_s = N_{xy}b^2/\pi^2 D_{11}$. The symbol κ is used for the eigenvalue because one matrix form of the problem defines $\lambda = \kappa^2$ for the eigenvalue. This matrix form follows from truncating the doubly infinite series in equations (99) and assigning single row subscripts to each pair of subscripts p and q. Thus, $w_{pq} = x_i$ when $p = p_i$ and $q = q_i$. Then, in the single subscript notation, equations (100) can be written in the standard form $(\mathbf{A} - \kappa\mathbf{B})\mathbf{X} = 0$. The matrix **A** is a diagonal matrix with $A_{ii} = k_{pq}$ for $p = p_i$ and $q = q_i$. The matrix **B** contains the coupling terms γ_{pqrs} defined in equations (101) and can be arranged in four submatrices. The four submatrices in matrix **B** are based on the form of the expression for γ_{pqrs}. Both $(p_i + r_j)$ and $(q_i + s_j)$ must be odd integers or $B_{ij} = -\gamma_{pqrs} = 0$. The sum of two odd integers is even; therefore, the sum $(p_i + q_i) + (r_j + s_j)$ is even or else $B_{ij} = 0$. For nonzero $B_{ij}, (p_i + q_i)$ and $(r_j + s_j)$ are both even or both odd. Therefore, the eigenproblem $(\mathbf{A} - \kappa\mathbf{B})\mathbf{X}$ can be written in partitioned form as

$$\begin{bmatrix} A_{ee} & 0 & 0 & 0 \\ 0 & A_{oo} & 0 & 0 \\ 0 & 0 & A_{eo} & 0 \\ 0 & 0 & 0 & A_{oe} \end{bmatrix} \begin{bmatrix} X_{ee} \\ X_{oo} \\ X_{eo} \\ X_{oe} \end{bmatrix} = \kappa \begin{bmatrix} 0 & B_{oo} & 0 & 0 \\ B_{ee} & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{oe} \\ 0 & 0 & B_{eo} & 0 \end{bmatrix} \begin{bmatrix} X_{ee} \\ X_{oo} \\ X_{eo} \\ X_{oe} \end{bmatrix} \tag{103}$$

where e and o indicate respectively even and odd values of the subscript integers. The partitioned problem decouples into two problems,

$$\begin{bmatrix} A_{ee} & 0 \\ 0 & A_{oo} \end{bmatrix} \begin{bmatrix} X_{ee} \\ X_{oo} \end{bmatrix} = \kappa \begin{bmatrix} 0 & B_{oo} \\ B_{ee} & 0 \end{bmatrix} \begin{bmatrix} X_{ee} \\ X_{oo} \end{bmatrix} \tag{104a}$$

and

$$\begin{bmatrix} A_{eo} & 0 \\ 0 & A_{oe} \end{bmatrix} \begin{bmatrix} X_{eo} \\ X_{oe} \end{bmatrix} = \kappa \begin{bmatrix} 0 & B_{oe} \\ B_{eo} & 0 \end{bmatrix} \begin{bmatrix} X_{eo} \\ X_{oe} \end{bmatrix} \tag{104b}$$

The first problem (eqs. 104a)) contains eigenvectors with components w_{pq} where $(p_i + q_i)$ is even. The mode shape w from equations (99) for this case is symmetric with respect to 180° rotations about an axis perpendicular to the plate at its centroid. The second problem (eqs. (104b)) contains modes that are antisymmetric and $(p_i + q_i)$ is odd.

The decoupled problems (eqs. (104)) are written as

$$X_{ee} = \kappa(A_{ee}^{-1} B_{oo}) X_{co} \tag{105a}$$

$$A_{oo} X_{oo} = \kappa^2 (B_{ee} A_{ee}^{-1} B_{oo}) X_{oo} \tag{105b}$$

and

$$X_{eo} = \kappa(A_{eo}^{-1} B_{oe}) X_{oe} \tag{106a}$$

$$A_{oe} X_{oe} = \kappa^2 (B_{eo} A_{eo}^{-1} B_{oe}) X_{oe} \tag{106b}$$

The method of delayed division for the SPLIT subroutine applied to equations (103) with $\lambda = \kappa$ failed to converge because the final 2×2 matrix in the forward sweep procedure did not contain any λ terms. However, the same subroutine with $\lambda = \kappa^2$ gives good results when applied to equations (105b) and (106b). The results from SPLIT for the isotropic case verify the results obtained when Stein and Neff used power iteration (ref. 6). Stein and Neff retained 10 Fourier coefficients for each mode in computing the lowest eigenvalue for equations (104a) and (104b). The results from SPLIT retaining more Fourier coefficients do not change the buckling loads significantly.

Another approach to solving equations (103) directly by delayed division is to use an eigenvalue shift. Let $\kappa = \bar{\kappa} + \lambda$, where $\bar{\kappa}$ is a nonzero approximate eigenvalue. Equations (103) are solved in the form $[(\mathbf{A} - \kappa\mathbf{B}) - \lambda\mathbf{B}]\mathbf{X} = 0$ with the SPLIT subroutine without any difficulty.

Table VI contains a test case obtained from use of an eigenvalue shift on equations (103). The case is an isotropic plate with $\beta = a/b = 2.0$. For this value of β, the lowest buckling load associated with an asymmetric mode is nearly coincident with the lowest buckling load associated with a symmetric mode. Although the buckling loads are the same to an accuracy of two digits, there is no problem sorting out the two mode shapes when delayed division is used.

TABLE VI. FOURIER COEFFICIENTS FOR SHEAR BUCKLING PROBLEM

$[\beta = a/b = 2.0]$

		Lowest symmetric mode w_{pq} from—		Lowest asymmetric mode w_{pq} from
p	q	SPLIT subroutine*	Reference 7†	SPLIT subroutine‡
1	1	1.0000	1.000	0
2	2	.3473	.343	0
3	1	−.3373	−.325	0
4	4	.0494	.048	0
4	2	−.0482	−.042	0
1	3	−.0318	−.032	0
5	1	−.0080	−.010	0
2	4	.0059	.005	0
3	5	.0036	.004	0
1	5	−.0019	−.002	0
2	1	0	0	1.0000
3	2	0	0	.3049
1	2	0	0	−.2790
4	1	0	0	−.1235
2	3	0	0	−.0707
4	3	0	0	.0318
5	2	0	0	.0092
3	4	0	0	.0044

*$K_S = 6.563$.
†$K_S = 6.59$.
‡$K_S = 6.610$.

The shear buckling problem illustrates a problem in which dropping nonlinear terms in the eigenvalue during the forward sweep can cause problems for the method of delayed division. Of the two approaches used to keep significant linear terms in the eigenvalue, the eigenvalue shift is the simplest to implement. The SPLIT subroutine only requires an input parameter to apply a shift.

Lagrangian Multipliers

The preceding example of shear buckling in plates illustrates the use of delayed division with an eigenvalue shift. The example in this section requires an eigenvalue shift and a new forward sweep with each iteration in the SPLIT subroutine. The eigenvalue shift with each iteration is essential because matrices **A** and **B** in the subroutine are not constant but are lambda matrices. A lambda matrix **A** is defined herein as a matrix whose elements are functions of a parameter λ:

$$A_{ij} = f_{ij}(\lambda) \tag{107}$$

A typical problem containing lambda matrices requires calculation of characteristic roots $\lambda = \lambda_m$ for which the determinant of the matrix **A** vanishes. Characteristic roots of lambda matrices can be computed with delayed division. Fewer iterations are required for each root when delayed division is used compared with the number of iterations for determinant-search methods. The problem $\mathbf{AX} = 0$ with $\mathbf{A}_{ij} = f_{ij}(\lambda)$ is rewritten for delayed division. The new problem is written with an iterative eigenvalue shift; convergence is achieved when the eigenvalue of the shifted problem approaches zero.

The eigenvalue shift is

$$\lambda_k = \lambda_{k-1} + \mu_k \qquad (k = 1, 2, ...) \tag{108a}$$

where k is an iteration counter. The general form $(\mathbf{A} - \lambda\mathbf{B})\mathbf{X} = 0$ for the method of delayed division is

$$A_{ij} = f_{ij}(\lambda_{k-1}) \tag{108b}$$

$$B_{ij} = -f'_{ij}(\lambda_{k-1}) \tag{108c}$$

$$\lambda = \mu_k \tag{108d}$$

where f'_{ij} denotes the derivative of f_{ij} with respect to λ. The smallest absolute value of the eigenvalue from the solution of $(\mathbf{A} - \lambda\mathbf{B})\mathbf{X}$ is taken as μ_k at each iteration step.

A subroutine, LAMR, that solves equations (108) is described in reference 1 along with example problems with lambda matrices. The SPLIT subroutine is used herein to take advantage of symmetry ($A_{ij} = A_{ji}$) in the example problem and to compute eigenvectors for multiple roots.

The lambda matrix in the example problem is generated from the application of the method of Lagrangian multipliers to a plate buckling problem. The method of Lagrangian multipliers reduces a linear eigenproblem with many degrees of freedom to a nonlinear eigenproblem with only a few degrees of freedom.

The particular problem treated herein is the buckling of an isotropic rectangular plate in compression and clamped on all four edges. An approximate solution for the clamped plate was obtained by Levy (ref. 8) in 1942. Levy also reported results of earlier investigations on the problem. Thurston (ref. 9) worked the problem using Lagrangian multipliers as a special case of a sandwich plate with infinite shear stiffness. The buckling loads in references 8 and 9 were computed with a determinant search method.

The analysis with Lagrangian multipliers starts when functions are found that satisfy the governing partial differential equation:

$$\nabla^4 w + \frac{-N_x}{D}\frac{\partial^2 w}{\partial x^2} = 0 \tag{109}$$

Consider functions of the form

$$\phi_{pq} = g_{pq}(x)\ \cos(q\pi y/b) \qquad \left\{\begin{array}{l}(p = 1, 2, ..., \infty) \\ (q = 0, 1, ..., \infty)\end{array}\right. \tag{110}$$

For each of the functions ϕ_{pq} which satisfy equation (109), the functions $g = g_{pq}(x)$ must satisfy the ordinary differential equation

$$\frac{d^4 g}{d\xi^4} - 2(q\beta\pi)^2\frac{d^2 g}{d\xi^2} + g(q\beta\pi)^4 + \frac{-N_x a^2}{D}\frac{d^2 g}{d\xi^2} = 0 \tag{111}$$

where $\xi = x/a$ and $\beta = a/b$. The solution of equation (111) is

$$g = c_1\cos(\alpha+\delta)\xi + c_2\sin(\alpha+\delta)\xi + c_3\cos(\alpha-\delta)\xi + c_4\sin(\alpha-\delta)\xi$$

where $\alpha^2 = K_{pq}(\pi\beta/2)^2$, $\delta^2 = \alpha^2 - (q\beta\pi)^2$, and

$$K_{pq} = \frac{-N_x b^2}{\pi^2 D} \tag{112}$$

The solution g contains four constants of integration. The constants of integration are determined so that each function $g_{pq}(x)$ satisfies the following boundary conditions at $x = 0$ and $x = a$:

$$g_{pq}(x) = 0 \tag{113a}$$

$$\frac{dg_{pq}}{dx} = 0 \tag{113b}$$

Since the boundary conditions (eqs. (113)) are homogeneous, the conditions determine a set of buckling coefficients K_{pq} and associated mode shapes with one arbitrary constant of integration.

After K_{pq} and the modal function $q_{pq}(x)$ are determined for each value of p and q, the next step in the Lagrangian multiplier method is to assume the solution for the clamped-plate problem is

$$w = \sum_{p=1}^{\infty}\sum_{q=0}^{\infty} w_{pq}g_{pq}(x)\cos(q\pi y/b) \tag{114}$$

where w_{pq} are constants to be determined. The solution w must satisfy the partial differential equation (109) plus the clamped-edge boundary conditions at $x = 0$ and $x = a$ and at $y = 0$ and $y = b$ of

$$w = 0 \tag{115a}$$

$$\frac{\partial w}{\partial x} = 0 \qquad (x = 0 \text{ or } x = a) \tag{115b}$$

$$\frac{\partial w}{\partial y} = 0 \qquad (y = 0 \text{ or } y = b) \tag{115c}$$

Since $g_{pq}(x)$ satisfies equations (113), each term of the series for w satisfies all the boundary conditions at the edges $x = 0$ and $x = a$. The cosine terms in y are chosen to satisfy equation (115c). The two conditions of zero deflection at the unloaded edges $y = 0$ and $y = b$ (eq. 115(a)) remain to be satisfied by the complete series

$$\sum_{p=1}^{\infty} \sum_{q=0}^{\infty} w_{pq} g_{pq}(x) = 0 \qquad (0 \leq x \leq a)$$

$$\sum_{p=1}^{\infty} \sum_{q=0}^{\infty} w_{pq} g_{pq}(x)(-1)^q = 0 \qquad (0 \leq x \leq a)$$

or, through addition and subtraction of equations, the problem decouples into symmetric and antisymmetric solutions as follows:

$$\sum_{p=1}^{\infty} \sum_{q=0,2}^{\infty} w_{pq} g_{pq}(x) = 0 \qquad (0 \leq x \leq a) \tag{116a}$$

$$\sum_{p=1}^{\infty} \sum_{q=1,3}^{\infty} w_{pq} g_{pq}(x) = 0 \qquad (0 \leq x \leq a) \tag{116b}$$

The coefficients w_{pq} must also be selected so that the series for w satisfies the partial differential equation (109) and gives

$$\nabla^4 w + \left(\frac{-N_x}{D}\right) \frac{\partial^2 w}{\partial x^2} = \frac{\pi^2}{b^2} \sum_{p=1}^{\infty} \sum_{q=0}^{\infty} w_{pq}(K_x - K_{pq}) \frac{d^2 g}{dx^2} \cos \frac{q\pi y}{b} = 0 \tag{117}$$

where $K_x = -N_x b^2/\pi^2 D$. The nondimensional buckling load coefficient K_x is the parameter λ in the lambda matrix for the clampled-plate problem. The lambda matrix is derived in the method of Lagrangian multipliers by use of a least-squares solution to reduce equations (116) and (117) to an algebraic problem.

The functions $g''_{p0}(x)$, the second derivatives of $g_{p0}(x)$, are a complete set of weighting functions for a least-squares solution of the boundary conditions (eqs. (116)). The boundary conditions become the algebraic constraint equations

$$\sum_{p=1}^{\infty} \sum_{q=0,2}^{\infty} w_{pq} L_{ipq} = 0 \qquad (i = 1, 2, ..., \infty) \tag{118a}$$

$$\sum_{p=1}^{\infty} \sum_{q=1,3}^{\infty} w_{pq} L_{ipq} = 0 \qquad (i = 1, 2, ..., \infty) \tag{118b}$$

The coefficients L_{ipq} in equations (118) are the definite integrals:

$$L_{ipq} = -l_i \int_0^a g''_{i0}(x)\, g_{pq}(x)\, dx = l_i \int_0^a g'_{i0}(x)\, g'_{pq}(x)\, dx \tag{119}$$

The constant l_i is a convenient normalizing factor:

$$\frac{1}{l_i} = \int_0^a [g'_{i0}(x)]^2\, dx$$

With the introduction of Lagrangian multipliers μ_i and γ_i, the algebraic problem is to minimize the function H:

$$H = \int_0^b \int_0^a w \left(\nabla^4 w + \frac{-N_x}{D} \frac{\partial^2 w}{\partial x^2} \right) dx\ dy - \sum_{i=1}^{\infty} \mu_i \left(\sum_{p=1}^{\infty} \sum_{q=0,2}^{\infty} w_{pq} L_{ipq} \right) - \sum_{i=1}^{\infty} \gamma_i \left(\sum_{p=1}^{\infty} \sum_{q=1,3}^{\infty} w_{pq} L_{ipq} \right) \tag{120}$$

The function H must be minimized with respect to the coefficients w_{pq} subject to the constraints in equations (118).

The reason for the choice of ϕ_{pq} (eqs. (110)) as coordinate functions now becomes apparent. The quadratic form in the double integral simplifies because ϕ_{pq} are orthonormal functions:

$$\int_0^b \int_0^a w(\nabla^4 w + \left(\frac{-N_x}{D} \frac{\partial^2 w}{\partial x^2} \right) dx\ dy = \frac{1}{2} \sum_{p=1}^{\infty} \sum_{q=0}^{\infty} w_{pq}^2 (K_{pq} - K_x) \tag{121}$$

The orthogonality of the set of functions

$$\phi_{pq} = g_{pq}(x)\ \cos(q\pi y/b)$$

follows directly from the relations

$$\int_0^b \cos(j\pi y/b)\ \cos(q\pi y/b)\ dy = 0 \qquad (j \neq q) \tag{122}$$

and

$$\int_0^a g_{iq} g_{pq}''\ dx = \int_0^a g_{iq}'' g_{pq}\ dx = -\int_0^a g_{iq}' g_{pq}'\ dx = 0 \qquad (i \neq p) \tag{123}$$

The functions ϕ_{pq} are made orthornormal by proper choice of the arbitrary constant of integration in each function $g_{pq}(x)$.

Substituting equations (121) into (120) and differentiating with respect to w_{pq} gives the following set of equations:

$$\frac{\partial H}{\partial w_{pq}} = 0$$

or

$$(K_{pq} - K_x) w_{pq} - \sum_{j=1}^{\infty} \mu_j L_{jpq} = 0 \qquad \left\{ \begin{array}{l} (p = 1, 2, ..., \infty) \\ (q = 0, 2, ..., \infty) \end{array} \right. \tag{124a}$$

$$(K_{pq} - K_x) w_{pq} - \sum_{j=1}^{\infty} \gamma_j L_{jpq} = 0 \qquad \left\{ \begin{array}{l} (p = 1, 2, ..., \infty) \\ (q = 1, 3, ..., \infty) \end{array} \right. \tag{124b}$$

Solving equations (124) for w_{pq} and eliminating w_{pq} from the constraints (eqs. (118)) gives the following two decoupled sets of equations:

$$\sum_{j=1}^{\infty} \left[\sum_{p=1}^{\infty} \sum_{q=0,2}^{\infty} \frac{L_{ipq} L_{jpq}}{(K_{pq} - K_x)} \right] \mu_j = 0 \qquad (i = 1, 2, ..., \infty) \tag{125a}$$

$$\sum_{j=1}^{\infty} \left[\sum_{p=1}^{\infty} \sum_{q=1,3}^{\infty} \frac{L_{ipq} L_{jpq}}{(K_{pq} - K_x)} \right] \gamma_j = 0 \qquad (i = 1, 2, ..., \infty) \tag{125b}$$

The matrix of coefficients of the Lagrangian multipliers μ_i and γ_i in equations (125) is a lambda matrix with $\lambda = K_x$, the buckling load coefficient for the clamped plate. Each element of the lambda matrix is a double infinite series rather than a polynomial in λ as in the classic literature of lambda matrices. The coefficients of the multipliers μ_i in equations (125a) are

$$A_{ij} = \sum_{p=1}^{\infty} \sum_{q=0,2}^{\infty} \frac{L_{ipq} L_{jpq}}{(K_{pq} - K_x)} \tag{126a}$$

A similar form appears in equations (125b). For nontrivial solutions of equations (125) the determinant of the lambda matrix must vanish. The iterative procedure in equations (108) is used in delayed division. Terms for A_{ij} in the form of equations (126a) correspond to equations (108b). The corresponding B_{ij} terms in equations (108c) are

$$B_{ij} = -\sum_{p=1}^{\infty} \sum_{q=0,2}^{\infty} \frac{L_{ipq} L_{jpq}}{(K_{pq} - K_x)^2} \tag{126b}$$

and $\lambda = \Delta K_x$ corresponds to the eigenvalue in equations (108d).

For numerical solutions, the double infinite set of constraint equations (125) must be truncated. Also, the double series in equations (126) must be truncated. Theory shows (e.g., refs. 10 to 12) that by careful truncation either upper or lower bounds on K_x can be computed numerically. Details of the theory are not explored herein except to note that the theory is useful in determining starting values for K_x in the iteration using equations (108).

The theory related to truncation provides guidance on the use of the SPLIT subroutine to compute the small eigenvalues $\lambda = \Delta K_x$ in the iteration sequence corresponding to equations (108). The forward sweep subroutine in SPLIT searches for large values of the coefficients A_{ij}. However, K_{pq} in the denominators of A_{ij} and B_{ij} in equations (126) leads to small values of A_{ij} associated with even smaller values of B_{ij}. To make the forward sweep subroutine eliminate higher modes first, the problem is "scaled." The scaling is carried out formally with the change in variables

$$x_i = (-B_{ii})^{-1/2} y_i \qquad (i = 1, 2, ..., n) \tag{127a}$$

so that $(\mathbf{A}^* - \lambda \mathbf{B}^*)\mathbf{Y} = 0$ is solved by the SPLIT subroutine for

$$A^*_{ij} = (-B_{ii})^{-1/2} A_{ij} (-B_{jj})^{-1/2} \tag{127b}$$

$$B^*_{ij} = (-B_{ii})^{-1/2} B_{ij} (-B_{jj})^{-1/2} \tag{127c}$$

There is no danger of taking square roots of negative numbers since $-B_{ii}$ in equation (127b) is a sum of squares of real numbers.

The matrix $\mathbf{B}^*$ has negative ones for diagonal elements. The matrix $\mathbf{A}^*$ has its largest absolute value elements on the diagonal, with small off-diagonal elements. The SPLIT subroutine picks out the smallest eigenvalue $\lambda_1 = \Delta K_x$ from equations (127) and the iteration indicated by equations (108) is continued by shifting K_x until λ_1 approaches zero. When $\lambda_1 = 0$ for equations (127), a critical value of K_x is determined for a nontrivial solution of equations (125). The eigenvectors of equations (127) determine the Lagrangian multipliers μ_i and γ_i, and equations (124) determine the coefficients w_{pq}. The complete deflection mode shape is determined with equation (114). The number of degrees of freedom (i.e., the total number of w_{pq} coefficients) can be an order of magnitude larger than the n constraint equations (125). The high accuracy of the final eigenvalues from a lambda matrix of relatively low order is the big advantage of the Lagrangian multiplier method. Table VII summarizes the results of K_x for the clamped plate with different number of degrees of freedom and with different numbers of constraints. The iteration sequence of equations (108) converges rapidly for all cases.

TABLE VII. BUCKLING COEFFICIENTS K_x FOR CLAMPED PLATE

(a) 20 constraints (eqs. 118); $w_{pq} = 1600$

	K_x from—	
$\beta = a/b$	Method of Lagrangian multipliers	Method of reference 8
0.25	64.0000	
.50	19.3402	
.75	11.6669	11.569
1.00	10.0755	10.074
1.25	9.2822	9.25
1.50	8.3697	8.33
1.75	8.1604	8.11
2.00	7.9335	7.88
3.00	7.4373	7.37
4.00	7.3001	7.23

(b) $\beta = a/b = 1.00$; Lagrangian multiplier method

Constraints	Maximum p	Maximum q	Total w_{pq}	K_x
4	400	20	20	9.8670
8	600	20	30	10.0211
20	1600	40	40	10.0755
20	400	20	20	10.1207
20	200	20	10	10.1234
20	120	20	6	10.1335
20	80	20	4	10.1594
8	100	10	10	10.1897
8	36	6	6	10.2580

The method of Lagrangian multipliers can be extended to more general problems. The solutions generated by program VIPASA for prismatic plate assemblies (refs. 13 and 14) satisfy periodic boundary conditions. The solutions with periodic boundary conditions obey orthogonality relations that give them a simple quadratic form similar to equations (121). Therefore, writing boundary conditions as constraint equations and using Lagrangian multipliers leads to lambda matrices similar to equations (125). The method of delayed division is a convenient tool for solving these equations for eigenvalues for the complete plate assembly under general boundary conditions.

Delayed Division and Other Methods

This report describes the method of delayed division for solving the symmetric eigenproblem $(\mathbf{A}-\lambda\mathbf{B})\mathbf{X} = 0$. Other methods exist for solving the eigenproblem. The SPLIT subroutine that was written to implement delayed division also incorporates some operations from other methods. The purpose of this section is to note the operations used in the SPLIT subroutine that are common to other methods. After the common operations in the SPLIT subroutine are noted, a brief discussion follows on modifications that can be made to the SPLIT subroutine to make it more efficient for certain cases. One case in which the SPLIT subroutine should be modified is when the number of eigenvalues m computed per forward sweep is relatively large. Another separate case in which the SPLIT subroutine should be modified is when **A** and **B** are sparse, banded matrices. The novel features in the method of delayed division are in the algorithm for reducing the problem $(\mathbf{A} - \lambda\mathbf{B})\mathbf{X} = 0$ to upper triangular form

$$(\mathbf{L}^{-1} + \lambda\mathbf{M})(\mathbf{A} - \lambda\mathbf{B})\mathbf{X} = (\mathbf{R} + \lambda\mathbf{S})\mathbf{X} = 0$$

In the SPLIT subroutine, the reduction to the triangular form is contained in the forward sweep subroutine and the back substitution subroutine. The exact computation $\mathbf{L}^{-1}\mathbf{A} = \mathbf{R}$ allows storage of information on the factors of the determinant of matrix **A**. The factors can be used for Sturm sequence calculations when **A** and **B** are constant matrices and **A** or **B** is positive definite (ref. 15). The factors can also be used when the matrix **A** is a lambda matrix that obeys a Sturm sequence (refs. 2, 12, and 13). In addition to factoring matrix **A**, the method of delayed division retains linear terms in λ. Therefore, because of the linear terms in λ in the factorization, delayed division converges faster than determinant-search methods. As a faster solution than determinant search, the input for the SPLIT subroutine is $m = 1$ or 2 followed by an eigenvalue shift after each forward sweep. The solution of the plate buckling problem by use of Lagrangian multipliers is a good example of the advantages of using the SPLIT subroutine rather than a determinant search.

The equivalence transformation $\mathbf{X} = \mathbf{QY}$ in the SPLIT subroutine contains characteristics borrowed from inverse iteration. Bathe and Wilson review the literature and tabulate operation counts for inverse iteration in references 16 to 18. When $m = 1$, that is, when only one eigenvalue and corresponding eigenvector are computed, the relation between delayed division and inverse iteration is clearest. The single approximate eigenvector identified by the forward sweep and back substitution appears in matrix **Q**. The transformed matrix $\mathbf{Q}^T(\mathbf{A} - \lambda\mathbf{B})\mathbf{Q}$ contains the two terms in the Rayleigh quotient for the approximate eigenvalue. The additional feature with delayed division is that the pivoting in the forward sweep automatically computes the approximate eigenvector associated with the lowest eigenvalue. No additional logic is required to start the iteration. Since delayed division retains linear terms in λ, the convergence of the inner loop containing the modified forward sweep and back substitution is faster than in simple inverse iteration. In the example problems of this report, acceptable engineering accuracy for the lowest eigenvalue and the associated eigenvector is obtained with one forward sweep and one modified sweep.

When the lowest eigenvalues are closely spaced, simple inverse iteration converges slowly. Delayed division does not have any problem with several small eigenvalues.

For several closely spaced, small eigenvalues, the version of inverse iteration that is appropriate is subspace iteration with Gram-Schmidt orthogonalization. The equivalence transformation $\mathbf{X} = \mathbf{QY}$ in the SPLIT subroutine has much in common with this version of inverse iteration. The number of lowest eigenvalues in the SPLIT subroutine m corresponds to the dimension of the subspace. The forward sweep in SPLIT automatically selects the m approximate eigenvectors. The retention of linear terms in λ speeds convergence.

The Gram-Schmidt orthogonalization in the SPLIT subroutine is satisfactory when m is small, say on the order of six. For larger values of m, the assumption that λ is small starts to break down in the sweep procedures and the Gram-Schmidt orthogonalization fails to preserve linear independence. The computer logic in the SPLIT subroutine needs to be modified after the equivalence transformation $\mathbf{X} = \mathbf{QY}$ to treat the case of large values of m. The transformed matrix $\mathbf{Q}^T(\mathbf{A} - \lambda\mathbf{B})\mathbf{Q}$ contains an $m \times m$ submatrix that can be solved exactly for all its eigenvalues and its eigenvectors which will be orthogonal over the subspace. A branch in the SPLIT subroutine with an exact solution over the subspace would be analogous to the method of inverse iteration with calculation of operator projections. Here again, as in the case for computing only one eigenvector, the logic in the forward sweep of the delayed division automatically computes the m vectors that span the subspace and speeds convergence.

The partitioning of the matrix $\mathbf{Q}^T(\mathbf{A}-\lambda\mathbf{B})\mathbf{Q}$ involved in the exact m dimensional subspace solution also has features similar to the Guyan reduction (ref. 19). Again the pivoting strategy in the forward sweep automates the reduction. Implicit in the Guyan reduction is the assumption that matrix **A** is positive definite. When **A** is positive definite, all the factors in its determinant are positive so that it is obvious that the pivoting in the forward sweep drives small eigenvalues of matrix **A** into the subspace. The retention of linear terms in λ in the forward sweep is a practical method of implementing the algorithm suggested by Kidder for modifying the Guyan reduction (ref. 20).

The partitioning suggested by the m dimensional subspace solution has already been programmed in the SPLIT subroutine. A branch solving the m dimensional subspace problem exactly should be simple to add.

The case for which **A** and **B** are large, sparse matrices requires a completely new subroutine in order to apply the method of delayed division. Pivoting is required to maintain numerical stability. Where **A** is positive definite, pivoting on the main diagonal is stable, does not increase the bandwidth of the reduced equations, and retains symmetry. Details of the pivoting strategy and storage requirements are beyond the scope of this report; however, the approach suggested by Gupta (ref. 15) for the storage problem is feasible for the algorithm in delayed division.

This discussion has shown that combining the advantages of the method of delayed division with other methods may be useful. As noted by many authors, the best combination of methods is problem dependent and there is no "best" solution to the eigenproblem $(\mathbf{A} - \lambda\mathbf{B})\mathbf{X} = 0$.

Concluding Remarks

The method of delayed division for efficiently solving the symmetric eigenproblem for small eigenvalues has been developed. Delayed division converges rapidly when the number of computed eigenvalues is small. The method computes the associated eigenvectors of the problem directly. The computation of the eigenvectors is numerically stable and has been programmed in a subroutine called SPLIT.

The SPLIT subroutine has been programmed to apply delayed division when the number of computed eigenvalues is small and **A** and **B** are dense matrices. The subroutine could be revised for cases with a larger number of computed eigenvalues. A new pivoting strategy is required before the method is efficient for large, sparse matrices.

NASA Langley Research Center
Hampton, VA 23665-5225
October 7, 1985

References

1. Thurston, G. A.: Roots of Lambda Matrices. Trans. ASME, *J. Appl. Mech.*, vol. 45, no. 4, Dec. 1978, pp. 859–863.
2. Anderson, Melvin S.: Buckling of Periodic Lattice Structures. *AIAA J.*, vol. 19, no. 6, June 1981, pp. 782–788.
3. Hurty, Walter C.; and Rubinstein, Moshe F.: *Dynamics of Structures.* Prentice-Hall, Inc., c.1964.
4. Wilkinson, J. H.: *The Algebraic Eigenvalue Problem.* Clarendon Press (Oxford), 1965.
5. Allen, D. N. de G.: *Relaxation Methods in Engineering and Science.* McGraw-Hill Book Co., Inc., c.1954.
6. Stein, Manuel; and Neff, John: *Buckling Stresses of Simply Supported Rectangular Flat Plates in Shear.* NACA TN 1222, 1947.
7. Batdorf, S. B.; and Stein, Manuel: *Critical Combinations of Shear and Direct Stress for Simply Supported Rectangular Flat Plates.* NACA TN 1223, 1947.
8. Levy, Samuel: Buckling of Rectangular Plates With Built-In Edges. *J. Appl. Mech.*, vol. 9, no. 4, Dec. 1942, pp. A-171–A-174.
9. Thurston, G. A.: Bending and Buckling of Clamped Sandwich Plates. *J. Aeronaut. Sci.*, vol. 24, no. 6, June 1957, pp. 407–412.
10. Courant, R.; and Hilbert, D.: *Methods of Mathematical Physics, Volume I.* Interscience Pub., Inc., c.1953.
11. Budiansky, Bernard; and Hu, Pai C.: *The Lagrangian Multiplier Method of Finding Upper and Lower Limits to Critical Stresses of Clamped Plates.* NACA Rep. 848, 1946. (Supersedes NACA TN 1103.)
12. Wittrick, W. H.; and Williams, F. W.: An Algorithm for Computing Critical Buckling Loads of Elastic Structures. *J. Struct. Mech.*, vol. 1, no. 4, 1973, 497–518.
13. Wittrick, W. H.; and Williams, F. W.: Buckling and Vibration of Anisotropic or Isotropic Plate Assemblies Under Combined Loadings. *Int. J. Mech. Sci.*, vol. 16, no. 4, Apr. 1974, pp. 209–239.
14. Anderson, Melvin S.; Stroud, W. Jefferson; Durling, Barbara J.; and Hennessy, Katherine W.: *PASCO: Structural Panel Analysis and Sizing Code, User's Manual.* NASA TM-80182, 1981. (Supersedes NASA TM-80182, 1980.)
15. Gupta, K. K.: *Solution of Eigenvalue Problems by Sturm Sequence Method. Int. J. Numer. Methods Eng.*, vol. 4, no. 3, May–June 1972, pp. 379–404.
16. Bathe, Klaus-Jürgen; and Wilson, Edward L.: Solution Methods for Eigenvalue Problems in Structural Mechanics. *Int. J. Numer. Methods Eng.*, vol. 6, no. 2, 1973, pp. 213–226.
17. Bathe, Klaus-Jürgen; and Wilson, Edward L.: *Numerical Methods in Finite Element Analysis.* Prentice-Hall, Inc., c.1976.
18. Bathe, Klaus Jürgen; and Wilson, Edward L.: Large Eigenvalue Problems in Dynamic Analysis. *J. Eng. Mech. Div.*, American Soc. Civil Eng., vol. 98, no. EM 6, Dec. 1972, pp. 1471–1485.
19. Guyan, Robert J.: Reduction of Stiffness and Mass Matrices. *AIAA J.*, vol. 3, no. 2, Feb. 1965, p. 380.
20. Kidder, Robert L.: Reduction of Structural Frequency Equations. *AIAA J.*, vol. 11, no. 6, June 1973, p. 892.

Symbols

$\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D},\mathbf{G},$ $\mathbf{L},\mathbf{M},\mathbf{R},\mathbf{S}$	coefficient matrices
$\mathbf{A}^*,\mathbf{B}^*$	scaled coefficient matrices
a	constant; plate length
$a_{ij}, b_{ij},$ $c_{ij}, d_{ij},$ g_{ij}	elements in ith row and jth column of coefficient matrices
b	constant; plate width
c	constant
D_{pq}	function in series solution for w in plate problem
EI	bending stiffness of column
g_{pq}	function of x in assumed form of solution for plate problem
g'_{pq}	derivative of g_{pq} with respect to ξ in plate problem
H	function to be minimized in Lagrangian multiplier method for plate problem
$\mathbf{I}$	identity matrix
K_{pq}	buckling coefficient for plates in compression
K_x, K_y	buckling coefficients for clamped plate
L	length of column
L_{ipq}	coefficient in plate problem
m	number of computed eigenvalues and corresponding eigenvectors
N	number of interior stations for column finite-difference equations
N_x	stress resultant in plate problem
n	total number of rows and columns in matrices $\mathbf{A}$ and $\mathbf{B}$
$O(\lambda^2)$	order of λ^2
P	axial load on column
$\mathbf{P}$	permutation matrix
P_E	Euler buckling load for simply supported column
p	constant
$\mathbf{Q}$	matrix in eigenvalue transformation
q	constant
$\mathbf{U}$	matrix of eigenvectors
u	component of $\mathbf{U}$
$\mathbf{V}$	matrix of approximate eigenvectors
v	component of $\mathbf{V}$
w	transverse deflection of plate
$\mathbf{X}$	vector
x	axial coordinate of column; Cartesian coordinate
$\mathbf{Y}$	vector in equivalence transformation
y	transverse deflection of column; Cartesian coordinate
α_i, β_i	nonzero elements of tridiagonal matrix
α, β, δ	constants in plate problem
Δ	determinant of 2×2 matrix
∇	biharmonic operator
κ	eigenvalue in plate problem
Λ	diagonal matrix containing eigenvalues
λ	eigenvalue
μ	shifted eigenvalue
μ_j, γ_j	Lagrange multipliers in plate problem
ξ	$= x/a$, nondimensional variable in plate problem
ϕ_{pq}	function in series solution for w in plate problem
Subscripts:	
I, J	constants in elimination of $\mathbf{X}$
$i, j, k, p,$ q, r, s	constants

Primes indicate matrices or elements after elementary operations in analysis or in computations.

1. Report No.
NASA TP-2514

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Solution of the Symmetric Eigenproblem $\mathbf{AX} = \lambda\mathbf{BX}$ by Delayed Division

5. Report Date
March 1986

6. Performing Organization Code
505-33-53-10

7. Author(s)
Gaylen A. Thurston and Nancy J. C. Bains

8. Performing Organization Report No.
L-15984

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, VA 23665-5225

10. Work Unit No.

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546-0001

13. Type of Report and Period Covered
Technical Paper

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Delayed division is an iterative method for solving the linear eigenvalue problem $\mathbf{AX} = \lambda\mathbf{BX}$ for a limited number of small eigenvalues and their corresponding eigenvectors. The distinctive feature of the method is the reduction of the problem to an approximate triangular form by systematically dropping quadratic terms in the eigenvalue λ. The report describes the pivoting strategy in the reduction and the method for preserving symmetry in submatrices at each reduction step. Along with the approximate triangular reduction, the report extends some techniques used in the method of inverse subspace iteration. Examples are included for problems of varying complexity.

17. Key Words (Suggested by Authors(s))
Eigenvalues
Symmetric
Eigenvectors

18. Distribution Statement
Unclassified—Unlimited

Subject Category 39

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	39	A03

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1986