INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 6, 213-226 (1973)

SOLUTION METHODS FOR EIGENVALUE PROBLEMS IN STRUCTURAL MECHANICS

KLAUS-JÜRGEN BATHE* AND EDWARD L. WILSON†
University of California, Berkeley, California, U.S.A.

SUMMARY

A survey of probably the most efficient solution methods currently in use for the problems $\mathbf{K}\boldsymbol{\phi} = \omega^2\mathbf{M}\boldsymbol{\phi}$ and $\mathbf{K}\boldsymbol{\psi} = \lambda\mathbf{K}_G\boldsymbol{\psi}$ is presented. In the eigenvalue problems the stiffness matrices $\mathbf{K}$ and $\mathbf{K}_G$ and the mass matrix $\mathbf{M}$ can be full or banded; the mass matrix can be diagonal with zero diagonal elements. The choice is between the well-known QR method, a generalized Jacobi iteration, a new determinant search technique and an automated subspace iteration. The system size, the bandwidth and the number of required eigenvalues and eigenvectors determine which method should be used on a particular problem. The numerical advantages of each solution technique, operation counts and storage requirements are given to establish guidelines for the selection of the appropriate algorithm. A large number of typical solution times are presented.

INTRODUCTION

In the dynamic response analysis of an assemblage of structural elements using conventional mode superposition the generalized eigenvalue problem

$$\mathbf{K}\boldsymbol{\phi} = \omega^2\mathbf{M}\boldsymbol{\phi} \tag{1}$$

is considered. In this equation $\mathbf{K}$ is the stiffness matrix and $\mathbf{M}$ is the mass matrix of the element assemblage, both are of order n.[7,20] The n solutions to equation (1) can be written as

$$\mathbf{K}\boldsymbol{\Phi} = \mathbf{M}\boldsymbol{\Phi}\boldsymbol{\Omega}^2 \tag{2}$$

where the columns in $\boldsymbol{\Phi}$ are the $\mathbf{M}$-orthonormalized eigenvectors (free vibration modes) $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n$ and $\boldsymbol{\Omega}^2$ is a diagonal matrix listing the eigenvalues $\omega_1^2, \ldots, \omega_n^2$ (free vibration frequencies squared).

Considering the complete dynamic analysis the most time consuming phase is usually the solution of the eigenvalue problem. For a most efficient solution it is necessary to take maximum advantage of the special properties of the matrices $\mathbf{K}$ and $\mathbf{M}$ and the specific solution requirements.

It is of particular importance that in structural analysis both matrices $\mathbf{K}$ and $\mathbf{M}$ are banded, i.e.

$$\left.\begin{aligned} k_{ij} &= 0 \quad \text{for } j > i + m_{\mathbf{K}} \\ m_{ij} &= 0 \quad \text{for } j > i + m_{\mathbf{M}} \end{aligned}\right\} \tag{3}$$

where $(2m_{\mathbf{K}}+1)$ and $(2m_{\mathbf{M}}+1)$ are the bandwidths of the matrices $\mathbf{K}$ and $\mathbf{M}$, respectively. Assuming that all rigid body modes have been removed from the system, $\mathbf{K}$ is positive definite. If in a finite element formulation a consistent mass matrix is used, $\mathbf{M}$ is also positive definite and $m_{\mathbf{M}} = m_{\mathbf{K}}$. In a lumped mass analysis $\mathbf{M}$ is diagonal with m_{ii} positive or zero.[7,20]

With regard to solution requirements it is usually not necessary to include in the mode superposition analysis the response in all modes. Many structures respond to particular types of dynamic loading primarily in a few modes, and the contribution of the other modes can be neglected. Also, the element assemblage must have been selected such that its lower frequencies and vibration mode shapes can accurately represent the structural response.[1] Therefore, in the solution of the eigenvalue problem we may reduce the numerical effort by only solving for the required lowest eigenvalues and corresponding eigenvectors.

---

* Assistant Research Engineer.

† Professor of Civil Engineering.

Received 24 June 1972
Revised 4 September 1972

Another generalized eigenvalue problem arises in buckling analysis. The equations governing buckling of an assemblage of structural elements are

$$\mathbf{K}\boldsymbol{\psi} = \lambda \mathbf{K}_G \boldsymbol{\psi} \tag{4}$$

where $\mathbf{K}$ is the small deflection stiffness matrix used in equation (1) and $\mathbf{K}_G$, which is always banded, is the geometric stiffness matrix of the element system. The eigenvalues give the buckling loads and the eigenvectors represent the corresponding buckling modes.[17] Because $\mathbf{K}_G$ is in general indefinite, equation (4) is re-written as

$$\mathbf{K}_G \boldsymbol{\psi} = \kappa \mathbf{K}\boldsymbol{\psi} \tag{5}$$

where $\kappa = 1/\lambda$ and can be negative or positive. In this equation the maximum value of κ is required (and possibly the next lowest values) which gives the lowest buckling load. Using a shift which is an upper bound on the maximum eigenvalue in equation (5), the problem is to determine the eigenvalue nearest to the shift.[2]

Many different solution procedures have been developed for eigenvalue problems in general, see Reference 19 for a list of references. More specifically of interest are the solution methods surveyed by Peters and Wilkinson[16] and Brönlund.[6] Reference should also be made to the work by Bauer,[5] Dong and others,[8] Jennings,[12] Jennings and Orr,[13] Gupta,[11] Felippa[10] and Rutishauser.[18] With a large number of different solution techniques available it need be noted that for the specific eigenvalue problems considered here, there is no single algorithm which always provides an efficient solution; however, it is only necessary to choose between a few most effective techniques.

The purpose of this paper is to summarize the probably most efficient solution techniques currently in use and to establish guidelines for the selection of the appropriate solution method for a given problem. The methods under consideration are the Householder–QR–inverse iteration technique,[19] a generalized Jacobi iteration,[2,9] a determinant search method[2,3] and an automated subspace iteration.[2,4] Only the basic steps of these solution methods are presented, where it is hoped that a structural analyst with relatively little experience in eigenvalue solution techniques can follow the exposition. The development of the individual techniques and their detailed relationships to other methods are given in the references.

The proper choice of solution method is most important in the analysis of large systems; however, the guidelines given are general and apply to the solution of any order eigenvalue problem. The numerical advantages of each of the solution methods are discussed. The high speed storage requirements and the number of operations needed for solution largely determine which of the methods is most efficient in specific practical problems. Typical solution times using the algorithms in a wide spectrum of practical analyses are presented in order to emphasize the recommendations given for their use.

As will be apparent later, there is little difficulty in choosing the appropriate algorithm in buckling analysis. For this reason, in the next sections specifically the solution of the problem $\mathbf{K}\boldsymbol{\phi} = \omega^2 \mathbf{M}\boldsymbol{\phi}$ is discussed; however, guidelines for the choice of algorithm in the solution of buckling problems also follow.

TRANSFORMATION OF GENERALIZED EIGENVALUE PROBLEM TO STANDARD EIGENVALUE PROBLEM

Much attention has been given to the solution of the standard eigenvalue problem.[19] The solution procedures developed can be used if the more general form of the eigenvalue problem

$$\mathbf{K}\boldsymbol{\phi} = \omega^2 \mathbf{M}\boldsymbol{\phi} \tag{6}$$

is first transformed to the standard form.

Assume that $\mathbf{M}$ is positive definite, then if $\mathbf{M} = \mathbf{S}\mathbf{S}^T$ for any non-singular matrix $\mathbf{S}$, the problem in equation (6) is equivalent to the solution of the standard eigenvalue problem

$$\bar{\mathbf{K}}\bar{\boldsymbol{\phi}} = \omega^2 \bar{\boldsymbol{\phi}} \tag{7}$$

where

$$\bar{\mathbf{K}} = \mathbf{S}^{-1}\mathbf{K}\mathbf{S}^{-T}, \quad \bar{\boldsymbol{\phi}} = \mathbf{S}^T \boldsymbol{\phi} \tag{8}$$

It is computationally efficient to use as $\mathbf{S}$ the Cholesky factor $\tilde{\mathbf{L}}_{\mathbf{M}}$ of $\mathbf{M}$, i.e. $\mathbf{M} = \tilde{\mathbf{L}}_{\mathbf{M}}\tilde{\mathbf{L}}_{\mathbf{M}}^{\mathrm{T}}$. The transformation is then a stable process provided $\mathbf{M}$ is well-conditioned with respect to inversion. However, if $\mathbf{M}$ is ill-conditioned the transformation process is also ill-conditioned; namely, as $\mathbf{M}$ becomes semi-definite, the system has very large eigenvalues and as $\omega_n^2 \leqslant \|\tilde{\mathbf{L}}_{\mathbf{M}}^{-1}\mathbf{K}\tilde{\mathbf{L}}_{\mathbf{M}}^{-\mathrm{T}}\|$† the elements in $\tilde{\mathbf{K}}$ are large and the eigenvalues of normal size are determined inaccurately.

Another transformation matrix $\mathbf{S}$ is obtained using the spectral decomposition of $\mathbf{M}$, i.e. $\mathbf{M} = \mathbf{R}\mathbf{D}^2\mathbf{R}^{\mathrm{T}}$, in which case $\mathbf{S} = \mathbf{R}\mathbf{D}$, where the columns in $\mathbf{R}$ are the eigenvectors and $\mathbf{D}^2$ is a diagonal matrix with the eigenvalues of $\mathbf{M}$. The use of this transformation matrix has an advantage because an ill-conditioning of $\mathbf{M}$ may now be concentrated in only a few small elements of $\mathbf{D}$. Then in $\tilde{\mathbf{K}}$ only those rows and columns corresponding to the small elements in $\mathbf{D}$ will have large elements and the eigenvalues of normal size are more likely to be preserved.

It is important to note that $\tilde{\mathbf{K}}$ has the same bandwidth as $\mathbf{K}$ when $\mathbf{M}$ is diagonal. In this case both transformation procedures give $\tilde{\mathbf{K}} = \mathbf{M}^{-\frac{1}{2}}\mathbf{K}\mathbf{M}^{-\frac{1}{2}}$, and the transformation is very cheap. However, consider that $\mathbf{M}$ is not a diagonal matrix; then the Cholesky transformation is still quite economical, but $\tilde{\mathbf{K}}$ is full, and if the order of the matrices is large, the solution of the standard eigenvalue problem in equation (7) can be very expensive.

It should also be pointed out that if $\mathbf{M}$ is ill-conditioned we may consider the problem $\mathbf{M}\boldsymbol{\phi} = (1/\omega^2)\mathbf{K}\boldsymbol{\phi}$ and use a decomposition of $\mathbf{K}$ instead. However, $\mathbf{K}$ is banded and therefore the transformation always leads to a full matrix.

STATIC CONDENSATION

The transformation of the generalized eigenvalue problem $\mathbf{K}\boldsymbol{\phi} = \omega^2\mathbf{M}\boldsymbol{\phi}$ to the standard eigenvalue problem $\tilde{\mathbf{K}}\tilde{\boldsymbol{\phi}} = \omega^2\tilde{\boldsymbol{\phi}}$ can only be carried out when $\mathbf{M}$ is positive definite. In lumped mass analysis $\mathbf{M}$ can have in general zero elements on the diagonal. In this case it is necessary to use first static condensation on the massless degrees of freedom.

Re-writing equation (6) as

$$\begin{bmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ac} \\ \mathbf{K}_{ca} & \mathbf{K}_{cc} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_a \\ \boldsymbol{\phi}_c \end{bmatrix} = \omega^2 \begin{bmatrix} \mathbf{M}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_a \\ \boldsymbol{\phi}_c \end{bmatrix} \tag{9}$$

where $\mathbf{M}_a$ is positive definite, we obtain the reduced generalized eigenvalue problem

$$\mathbf{K}_a\boldsymbol{\phi}_a = \omega^2\mathbf{M}_a\boldsymbol{\phi}_a \tag{10}$$

where

$$\mathbf{K}_a = \mathbf{K}_{aa} - \mathbf{K}_{ac}\mathbf{K}_{cc}^{-1}\mathbf{K}_{ca} \tag{11}$$

and

$$\boldsymbol{\phi}_c = -\mathbf{K}_{cc}^{-1}\mathbf{K}_{ca}\boldsymbol{\phi}_a \tag{12}$$

In practice $\mathbf{K}_a$ can be obtained as follows

$$\mathbf{K}_{cc} = \tilde{\mathbf{L}}_c\tilde{\mathbf{L}}_c^{\mathrm{T}}, \quad \tilde{\mathbf{L}}_c\mathbf{Y} = \mathbf{K}_{ca}, \quad \mathbf{K}_a = \mathbf{K}_{aa} - \mathbf{Y}^{\mathrm{T}}\mathbf{Y} \tag{13}$$

where $\tilde{\mathbf{L}}_c$ is the Cholesky factor of $\mathbf{K}_{cc}$.

Instead of using equation (10), alternatively, a flexibility matrix $\mathbf{F}_a$ corresponding to the mass degrees of freedom, i.e. $\mathbf{F}_a = \mathbf{K}_a^{-1}$, could be calculated.[2] The eigenvalue problem then to be considered is $(1/\omega^2)\boldsymbol{\phi}_a = \mathbf{F}_a\mathbf{M}_a\boldsymbol{\phi}_a$, which using a factorization of $\mathbf{M}_a$ can obviously also be transformed to the standard form.

Although the order of the matrices in the eigenvalue problem has been reduced, matrix $\mathbf{K}_a$ (and certainly $\mathbf{F}_a$) is in general full. To decrease computational requirements in the solution of equation (10) the mass of

† $\|\ \|$ denotes any norm.

the structure may have been lumped at only a few degrees of freedom. This can be appropriate in the analysis of some structures, such as high-rise buildings. However, depending on the engineer's experience, in the analysis of complex structures the calculated eigensystem may then only be a very crude approximation to the required eigensystem of the actual structure.

HOUSEHOLDER–QR–INVERSE ITERATION SOLUTION

A very efficient procedure, which is probably regarded as the best method for finding the complete eigensystem of $\tilde{\mathbf{K}}$ in equation (7), is the Householder–QR–inverse iteration solution.[19] The name suggests the following three solution steps:

1. Householder transformations are used to reduce the matrix to tridiagonal form.
2. QR iteration yields the eigenvalues.
3. Using inverse iteration the eigenvectors of the tridiagonal matrix are calculated and transformed to the eigenvectors of $\tilde{\mathbf{K}}$.

The Householder reduction

The Householder reduction to tridiagonal form involves $(n-2)$ orthogonal similarity transformations

$$\tilde{\mathbf{K}}_{k+1} = \mathbf{P}_k^{\mathrm{T}} \tilde{\mathbf{K}}_k \mathbf{P}_k, \quad k = 1, 2, \ldots, n-2, \quad \tilde{\mathbf{K}}_1 = \tilde{\mathbf{K}} \tag{14}$$

where

$$\mathbf{P}_k = \mathbf{I} - \theta \mathbf{w}_k \mathbf{w}_k^{\mathrm{T}} \tag{15}$$

$$\theta = \frac{2}{\mathbf{w}_k^{\mathrm{T}} \mathbf{w}_k} \tag{16}$$

Consider the case $k = 1$, which is typical. Let

$$\mathbf{P}_1 = \left[\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0} & \bar{\mathbf{P}}_1 \end{array}\right], \quad \mathbf{w}_1 = \left[\begin{array}{c} 0 \\ \hline \bar{\mathbf{w}}_1 \end{array}\right]$$

and

$$\tilde{\mathbf{K}}_1 = \left[\begin{array}{c|c} \tilde{k}_{11} & \tilde{\mathbf{k}}_1^{\mathrm{T}} \\ \hline \tilde{\mathbf{k}}_1 & \tilde{\mathbf{K}}_{11} \end{array}\right]$$

Then

$$\tilde{\mathbf{K}}_2 = \left[\begin{array}{c|c} \tilde{k}_{11} & \tilde{\mathbf{k}}_1^{\mathrm{T}} \bar{\mathbf{P}}_1 \\ \hline \bar{\mathbf{P}}_1^{\mathrm{T}} \tilde{\mathbf{k}}_1 & \bar{\mathbf{P}}_1^{\mathrm{T}} \tilde{\mathbf{K}}_{11} \bar{\mathbf{P}}_1 \end{array}\right] \tag{17}$$

The vector $\bar{\mathbf{w}}_1$ is determined from the condition

$$(\mathbf{I} - \theta \bar{\mathbf{w}}_1 \bar{\mathbf{w}}_1^{\mathrm{T}}) \tilde{\mathbf{k}}_1 = \pm \|\tilde{\mathbf{k}}_1\|_2 \mathbf{e}_1 \dagger \tag{18}$$

where $\mathbf{e}_1$ is the $(n-1)$ dimensional unit vector, i.e., $\mathbf{e}_1^{\mathrm{T}} = [1\ 0\ 0\ \ldots\ 0]$. It is only necessary to solve from equation (18) for a multiple of $\bar{\mathbf{w}}_1$, and we can use

$$\bar{\mathbf{w}}_1 = \tilde{\mathbf{k}}_1 + \mathrm{sign}(\tilde{k}_{21}) \|\tilde{\mathbf{k}}_1\|_2 \mathbf{e}_1 \tag{19}$$

The equivalent steps for $k = 2, 3, \ldots, n-2$ are obvious.

In the calculations we can use the symmetry property of $\tilde{\mathbf{K}}$ and store only the lower triangular part of the matrix. Also, we can use the storage locations of the elements which are zeroed in the reduction in order to store the $\mathbf{w}_k$ for the calculation of the eigenvectors.

---

† $\|\ \|_2$ denotes Euclidean norm.

The QR iteration

Consider now the QR iteration with shifts on the tridiagonal matrix $\tilde{\mathbf{K}}_{n-1}$, which we call $\mathbf{T}_1$. The iteration is as follows

$$\mathbf{T}_k - \mu_k \mathbf{I} = \mathbf{Q}_k \mathbf{R}_k \tag{20}$$

$$\mathbf{T}_{k+1} = \mathbf{R}_k \mathbf{Q}_k + \mu_k \mathbf{I}, \quad k = 1, 2, \ldots \tag{21}$$

where $\mathbf{Q}_k$ is an orthogonal matrix, $\mathbf{R}_k$ is an upper triangular matrix and μ_k is the shift. In each iteration we perform an orthogonal similarity transformation

$$\mathbf{T}_{k+1} = \mathbf{Q}_k^{\mathrm{T}} \mathbf{T}_k \mathbf{Q}_k \tag{22}$$

and then

$$\mathbf{T}_{k+1} \to \boldsymbol{\Omega}^2 \quad \text{as } k \to \infty$$

Regarding the convergence of the iteration it can be shown that the QR iteration is intimately related to the probably more familiar inverse iteration.[19] In particular, the QR iteration with μ_k properly chosen corresponds to the Rayleigh quotient iteration, which converges cubically in the neighbourhood of an eigenvalue.[15] In the iteration the eigenvalues are not found in order of their magnitudes and it is usual practice to calculate them all. Ortega and Kaiser[14] have developed explicit formulae which relate the elements in $\mathbf{T}_{k+1}$ to the elements in $\mathbf{T}_k$.

Solution of eigenvectors

Once the eigenvalues have been obtained to full machine precision we calculate only the required eigenvectors of $\mathbf{T}_1$ by simple inverse iteration with shifts equal to the corresponding eigenvalues. Two steps of inverse iteration are usually sufficient. These vectors need be transformed with the Householder transformations used to obtain the eigenvectors of $\tilde{\mathbf{K}}$.

Table I summarizes the Householder–QR–inverse iteration algorithm and gives the high speed storage and number of operations required for solution. In the operation counts one operation is assumed to consist of one multiplication which nearly always is followed by an addition.

Table I. Summary of Householder–QR–inverse iteration solution

Operation	Calculation	Number of operations	Required storage
Householder transformations	$\tilde{\mathbf{K}}_{k+1} = \mathbf{P}_k^{\mathrm{T}} \tilde{\mathbf{K}}_k \mathbf{P}_k, \quad k = 1, 2, \ldots, n-2$ $\tilde{\mathbf{K}}_1 = \tilde{\mathbf{K}}$	$\frac{2}{3}n^3 + \frac{3}{2}n^2$	
QR iterations	$\mathbf{T}_{k+1} = \mathbf{Q}_k^{\mathrm{T}} \mathbf{T}_k \mathbf{Q}_k, \quad k = 1, 2, \ldots$ $\mathbf{T}_1 = \tilde{\mathbf{K}}_{n-1}$	$9n^2$	Using symmetry of matrix
Calculation of p eigenvectors	$(\tilde{\mathbf{K}}_{n-1} - \omega_i^2 \mathbf{I})\, \mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)}, \quad k = 1, 2$ $i = 1, 2, \ldots, p$	$10\,pn$	$\frac{1}{2}[n(n+1)] + 6n$
Transformation of eigenvectors	$\tilde{\boldsymbol{\phi}}_i = \mathbf{P}_1 \ldots \mathbf{P}_{n-2}\, \mathbf{x}_i^{(3)}, \quad i = 1, 2, \ldots, p$	$pn(n-1)$	
Total for all eigenvalues and p eigenvectors		$\frac{2}{3}n^3 + \frac{21}{2}n^2 + pn^2 + 9\,pn$	

As was noted above, the complete solution of the generalized eigenvalue problem requires static condensation of the massless degrees of freedom and the transformation to the standard eigenvalue problem. The storage requirements and operations for these calculations are not included in the table.

It should be noted that this operation count as well as those given in the next sections represents an estimate of the actual number of operations performed by a solution routine. Only the significant terms are included in the operation counts and the actual number of operations will vary slightly depending on programming details.

GENERALIZED JACOBI ITERATION SOLUTION

For solution of the generalized eigenvalue problem using the Householder–QR–inverse iteration technique, we first need to transform the problem to the standard form. A generalized Jacobi iteration avoids this transformation and solves directly for all eigenvalues and eigenvectors of the generalized eigenvalue problem. The massless degrees of freedom in a lumped mass analysis do not need to be statically condensed. When the off-diagonal elements are small or sparse this direct solution can be very efficient. Another advantage of the method is that the algorithm is very simple.[2,9]

In the iteration let $\mathbf{K}_1 = \mathbf{K}$ and $\mathbf{M}_1 = \mathbf{M}$, then we form for $k = 1, 2, \ldots$

$$\mathbf{K}_{k+1} = \mathbf{P}_k^{\mathrm{T}} \mathbf{K}_k \mathbf{P}_k, \quad \mathbf{M}_{k+1} = \mathbf{P}_k^{\mathrm{T}} \mathbf{M}_k \mathbf{P}_k \tag{23}$$

where $\mathbf{P}_k$ is a generalized rotation matrix

$$\mathbf{P}_k = \begin{array}{c} \begin{matrix} & & i & j & \\ \end{matrix} \\ \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & \alpha & \\ & & \gamma & 1 & \\ & & & & \ddots \\ & & & & & 1 \end{bmatrix} \end{array} \begin{matrix} \\ \\ \\ \leftarrow i \\ \leftarrow j \\ \\ \end{matrix} \tag{24}$$

The variables α and γ are selected to zero simultaneously the (i,j) elements in $\mathbf{K}_k$ and $\mathbf{M}_k$, where then assuming that $\mathbf{M}$ is positive definite

$$\mathbf{K}_{k+1} \to \mathrm{diag}\,(\mathscr{K}_r), \quad \mathbf{M}_{k+1} \to \mathrm{diag}\,(\mathscr{M}_r) \quad \text{as } k \to \infty$$

The required eigenvalues are

$$\boldsymbol{\Omega}^2 = \mathrm{diag}\,(\mathscr{K}_r / \mathscr{M}_r) \tag{25}$$

and if l iterations were required for convergence the required eigenvectors are

$$\boldsymbol{\Phi} = \mathbf{P}_1 \mathbf{P}_2 \ldots \mathbf{P}_l \,\mathrm{diag}\,(\mathscr{M}_r^{-\frac{1}{2}}) \tag{26}$$

When $\mathbf{M}$ is diagonal with zero diagonal elements simple provisions need be made for it in the algorithm.

The generalized Jacobi iteration has been found to be very efficient when implemented as a threshold iteration in which a rotation is applied if either of the coupling factors

$$\frac{k_{ij}^{(k)^2}}{k_{ii}^{(k)} k_{jj}^{(k)}} \quad \text{and} \quad \frac{m_{ij}^{(k)^2}}{m_{ii}^{(k)} m_{jj}^{(k)}} \tag{27}$$

where the superscript refers to the matrices $\mathbf{K}_k$ and $\mathbf{M}_k$, is larger than the current threshold. Naturally, when $\mathbf{M}$ is diagonal the mass coupling factor is always set equal to zero.

The calculations in general performed in the solution are summarized in Table II, where the storage requirements and an operation count are also given. The total number of operations in one sweep are an upper bound because it is assumed that both matrices are full and that all off-diagonal elements are zeroed, i.e. the threshold tolerance is never passed. The actual number of operations per sweep is naturally much less if the off-diagonal elements are already small or the matrices are not full. The total number of sweeps for solution depends on the magnitude and positioning of the off-diagonal elements and the eigensystem accuracy required. In general, the solution can be obtained in two to eight sweeps.

DETERMINANT SEARCH SOLUTION

In the solution of large eigenvalue problems the number of required eigenvalues and corresponding vectors is usually much smaller than the order of the matrices. In this case it is much more economical to find only the required eigensystem.

Table II. Summary of generalized Jacobi solution

Operation	Calculation	Number of operations	Required storage
Calculation of coupling factors	$\frac{k_{ij}^{(k)2}}{k_{ii}^{(k)}\,k_{jj}^{(k)}},\quad \frac{m_{ij}^{(k)2}}{m_{ii}^{(k)}\,m_{jj}^{(k)}}$	6	
Transformation to zero elements (i,j)	$\bar{k}_{ii}^{(k)} = k_{ii}^{(k)}\,m_{ij}^{(k)} - m_{ii}^{(k)}\,k_{ij}^{(k)}$ $\bar{k}_{jj}^{(k)} = k_{jj}^{(k)}\,m_{ij}^{(k)} - m_{jj}^{(k)}\,k_{ij}^{(k)}$ $\bar{k}^{(k)} = k_{ii}^{(k)}\,m_{jj}^{(k)} - k_{jj}^{(k)}\,m_{ii}^{(k)}$ $x = \frac{\bar{k}^{(k)}}{2} + (\operatorname{sign}\bar{k}^{(k)})\sqrt{\left[\left(\frac{\bar{k}^{(k)}}{2}\right)^2 + \bar{k}_{ii}^{(k)}\bar{k}_{jj}^{(k)}\right]}$ $\gamma = -\frac{\bar{k}_{ii}^{(k)}}{x},\quad \alpha = \frac{\bar{k}_{jj}^{(k)}}{x}$ $\mathbf{K}_{k+1} = \mathbf{P}_k^{\mathrm{T}}\mathbf{K}_k\mathbf{P}_k,\quad \mathbf{M}_{k+1} = \mathbf{P}_k^{\mathrm{T}}\mathbf{M}_k\mathbf{P}_k$	$4n+12$	Using symmetry of matrices $n(n+2)$
Calculation of eigenvectors	$(\mathbf{P}_1 \ldots \mathbf{P}_{k-1})\,\mathbf{P}_k$	$2n$	n^2
	Total for one sweep	$3n^3+6n^2$	$2n^2+2n$

For matrices with small bandwidth a determinant search algorithm provides a very efficient solution.[2,3] The algorithm uses triangular factorization and vector inverse iteration directly on the general problem $\mathbf{K}\boldsymbol{\phi} = \omega^2\mathbf{M}\boldsymbol{\phi}$ and solves for the required eigenvalues and vectors in succession from the leastdom inant eigenpair upwards. In the eigenvalue problem $\mathbf{M}$ can be diagonal, with zero diagonal elements, or may be banded positive definite.

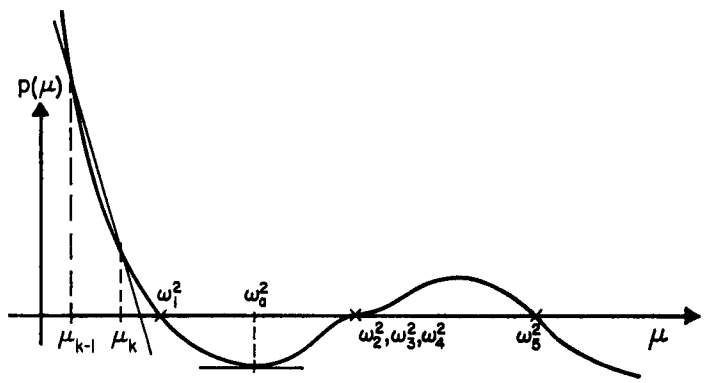

Figure 1. Characteristic polynomial $p(\mu)$

Consider the solution for the eigenpair $(\omega_1^2, \boldsymbol{\phi}_1)$, where ω_1^2 may be a multiple root.

The first objective in the iteration is to obtain a shift near ω_1^2. Here we use the fact that the eigenvalues are the roots of the characteristic polynomial $p(\mu) = \det(\mathbf{K}-\mu\mathbf{M})$. To evaluate $p(\mu)$ the matrix $\mathbf{K}-\mu\mathbf{M}$ is factored into $\mathbf{LDL}^{\mathrm{T}}$ using Gauss elimination, where $\mathbf{L}$ is a unit lower triangular matrix and $\mathbf{D}$ is a diagonal matrix. We then have $p(\mu) = \Pi_{i=1}^{n} d_{ii}$.

Let $\mu_{k-1} < \mu_k < \omega_1^2$ as shown in Figure 1. The next shift μ_{k+1} is calculated using an accelerated secant iteration in which

$$\mu_{k+1} = \mu_k + \eta\,\frac{p(\mu_k)}{p(\mu_k) - p(\mu_{k-1})}\,(\mu_{k-1} - \mu_k) \tag{28}$$

where η is a constant. When $\eta = 1{\cdot}0$ we have the well-known secant iteration in which case $\mu_{k+1} \leqslant \omega_1^2$ and $\mu_{k+1} \to \omega_1^2$ as $k \to \infty$. However, convergence in this iteration can be slow. Because the aim is to obtain merely

a shift near ω_1^2 the program uses an efficient acceleration scheme in which $\eta \geqslant 2{\cdot}0$. Starting the iteration η equals 2·0 because in this case $\mu_{k+1} \leqslant \omega_a^2$, where ω_a^2 is the smallest stationary point of p. A jump over a simple root would be detected by a sign change in p. However, when we iterate towards a multiple root or a cluster of roots, convergence with $\eta = 2{\cdot}0$ is still slow. Fortunately, in this case the eigenvalue separation theorem (Sturm sequence property) allows us to accelerate the iteration further by increasing η still more.

Once a shift near ω_1^2 has been obtained by either jumping over it or by approaching it sufficiently close from below, inverse iteration is used to calculate the eigenvector $\boldsymbol{\phi}_1$ and the Rayleigh correction ρ^c, which added to the shift gives the eigenvalue to the required precision (see Table III).

Table III. Summary of determinant search solution

Operation	Calculation	Number of operations $m = m_{\mathbf{K}} = m_{\mathbf{M}}$	Number of operations $m = m_{\mathbf{K}}, m_{\mathbf{M}} = 0$
Secant iteration	$\bar{\mathbf{K}} = \mathbf{K} - \mu_k \mathbf{M}$	$n(m+1)$	n
	$\bar{\mathbf{K}} = \mathbf{LDL}^{\mathrm{T}}$	$\frac{1}{2}nm^2 + \frac{3}{2}nm$	$\frac{1}{2}nm^2 + \frac{3}{2}nm$
	$\mathrm{p}(\mu_k) = \prod_{i=1}^{n} d_{ii}$	n	n
Inverse iteration	$\bar{\mathbf{K}}\bar{\mathbf{x}}_{k+1} = \mathbf{y}_k$	$n(2m+1)$	$n(2m+1)$
	$\bar{\mathbf{y}}_{k+1} = \mathbf{M}\bar{\mathbf{x}}_{k+1}$	$n(2m+1)$	n
	$\rho^c(\bar{\mathbf{x}}_{k+1}) = \dfrac{\bar{\mathbf{x}}_{k+1}^{\mathrm{T}} \mathbf{y}_k}{\bar{\mathbf{x}}_{k+1}^{\mathrm{T}} \bar{\mathbf{y}}_{k+1}}$	$2n$	$2n$
	$\mathbf{y}_{k+1} = \left(\bar{\mathbf{y}}_{k+1} - \sum_{j=1}^{6} \alpha_{i-j} \bar{\boldsymbol{\phi}}_{i-j}\right) \Big/ (\bar{\mathbf{x}}_{k+1}^{\mathrm{T}} \bar{\mathbf{y}}_{k+1})^{\frac{1}{2}}$ where $\bar{\boldsymbol{\phi}}_j = \mathbf{M}\boldsymbol{\phi}_j$, $\alpha_j = \bar{\mathbf{x}}_{k+1}^{\mathrm{T}} \bar{\boldsymbol{\phi}}_j$	$13n$	$13n$
Total for p lowest eigenvalues and associated eigenvectors assuming six secant and six inverse iterations per eigenpair		$(3nm^2 + 39nm + 114n)p$	$(3nm^2 + 21nm + 114n)p$

Required storage $m = m_{\mathbf{K}} = m_{\mathbf{M}}$	Required storage $m = m_{\mathbf{K}}, m_{\mathbf{M}} = 0$
Using symmetry of matrices	
$2n(m+1) + 9n$	$n(m+1) + 10n$

This iteration for ω_1^2 and $\boldsymbol{\phi}_1$ is typical because the advantage of the one-sided approach to ω_1^2 is also obtained for any other root, say ω_{j+1}^2 by using instead of $\mathrm{p}(\mu)$ in equation (28) the deflated polynomial $\mathrm{p}_j(\mu)$, Figure 2, where

$$\mathrm{p}_j(\mu) = \mathrm{p}(\mu) \Big/ \prod_{i=1}^{j} (\mu - \omega_i^2) \tag{29}$$

The calculations in a secant iteration and in a vector inverse iteration are summarized in Table III, where also the required number of operations and the storage requirements are given. Note that the factorization of $\mathbf{K} - \mu\mathbf{M}$ is performed without interchanges which has proven to be numerically adequate.[2] Also, the iteration vector is orthogonalized in each iteration to the last found six eigenvectors. In this operation count the half bandwidths $m_{\mathbf{M}}$ and $m_{\mathbf{K}}$ are assumed to be full, and terms involving the bandwidths only have been neglected. In most actual systems the bandwidths vary and many zeros occur within the band. The solution routine should take due account of both. The number of iterations required for the solution of an eigenpair depends on the system under consideration; experience shows that about six secant steps and six inverse iterations are required.

The determinant search technique is most efficient and has been implemented as an in-core solution routine. Because relatively many triangular factorizations are required, much tape handling would be necessary in an out-of-core solution.[2] Also, the technique is most efficient in the analysis of small-banded systems, and in this case relatively large order systems can be solved on reasonable size computers.

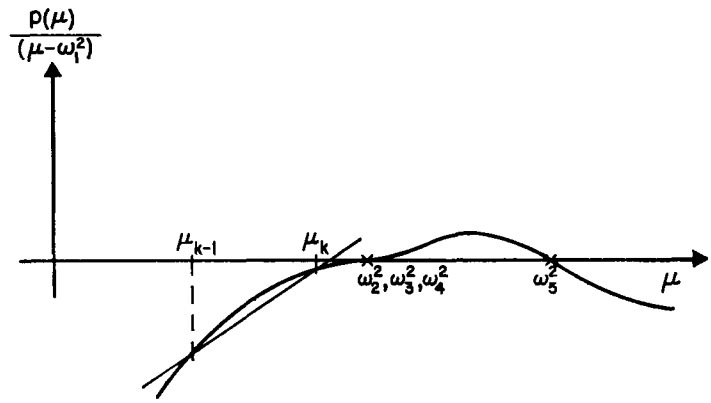

Figure 2. p(μ) with ω_1^2 suppressed

SUBSPACE ITERATION SOLUTION

In the subspace iteration solution the required eigenvalues and vectors are also calculated directly without a transformation to the standard form.[2,4]

The aim is to solve for the p lowest eigenvalues and associated eigenvectors satisfying

$$\mathbf{K}\boldsymbol{\Phi} = \mathbf{M}\boldsymbol{\Phi}\boldsymbol{\Omega}^2 \tag{30}$$

where the columns in $\boldsymbol{\Phi}$ are the p eigenvectors and $\boldsymbol{\Omega}^2$ is a diagonal matrix with the corresponding eigenvalues. The specific idea used in the solution is that the eigenvectors form an **M**-orthonormal basis of the p-dimensional least dominant subspace of the operators **K** and **M**.

In the solution we iterate simultaneously with q linearly independent vectors, where $q \geq p$. In the kth iteration the vectors span the q-dimensional subspace $\mathscr{E}_{k+1}$ and 'best' eigenvalue and eigenvector approximations are calculated, i.e. when the vectors span the p-dimensional least dominant subspace the required eigenvalues and eigenvectors are obtained.

Let $\mathbf{X}_1$ store the starting vectors, then the algorithm is defined as follows:

For $k = 1, 2, \ldots$ iterate from $\mathscr{E}_k$ to $\mathscr{E}_{k+1}$

$$\mathbf{K}\bar{\mathbf{X}}_{k+1} = \mathbf{M}\mathbf{X}_k \tag{31}$$

Find the projections of the operators **K** and **M** onto $\mathscr{E}_{k+1}$

$$\mathbf{K}_{k+1} = \bar{\mathbf{X}}_{k+1}^{\mathrm{T}}\mathbf{K}\bar{\mathbf{X}}_{k+1} \tag{32}$$

$$\mathbf{M}_{k+1} = \bar{\mathbf{X}}_{k+1}^{\mathrm{T}}\mathbf{M}\bar{\mathbf{X}}_{k+1} \tag{33}$$

Solve for the eigensystem of the projected operators

$$\mathbf{K}_{k+1}\mathbf{Q}_{k+1} = \mathbf{M}_{k+1}\mathbf{Q}_{k+1}\boldsymbol{\Omega}_{k+1}^2 \tag{34}$$

Find an improved approximation to the eigenvectors

$$\mathbf{X}_{k+1} = \bar{\mathbf{X}}_{k+1}\mathbf{Q}_{k+1} \tag{35}$$

Then provided the starting subspace is not orthogonal to one of the required eigenvectors, we have

$$\boldsymbol{\Omega}_{k+1}^2 \to \boldsymbol{\Omega}^2, \quad \mathbf{X}_{k+1} \to \boldsymbol{\Phi} \quad \text{as } k \to \infty$$

The iteration is performed with q vectors because the asymptotic convergence rate of the ith column in $\mathbf{X}_{k+1}$ to $\boldsymbol{\phi}_i$ is given by $\omega_i^2/\omega_{q+1}^2$; therefore, the larger q the higher the convergence rate for the p vectors of interest, but also more operations need be carried out in each iteration. In the implementation of the algorithm $q = \min\{2p, p+8\}$ has been found to be effective.

Table IV. Summary of subspace iteration solution

Operation	Calculation	Number of operations $m = m_{\mathbf{K}} = m_{\mathbf{M}}$	Number of operations $m = m_{\mathbf{K}}, m_{\mathbf{M}} = 0$	Required storage
Factorization of $\mathbf{K}$	$\mathbf{K} = \mathbf{LDL}^{\mathrm{T}}$	$\frac{1}{2}nm^2 + \frac{3}{2}nm$	$\frac{1}{2}nm^2 + \frac{3}{2}nm$	
Subspace iteration	$\mathbf{K}\bar{\mathbf{X}}_{k+1} = \mathbf{Y}_k$	$nq(2m+1)$	$nq(2m+1)$	
	$\mathbf{K}_{k+1} = \bar{\mathbf{X}}_{k+1}^{\mathrm{T}}\mathbf{Y}_k$	$\frac{1}{2}nq(q+1)$	$\frac{1}{2}nq(q+1)$	
	$\bar{\mathbf{Y}}_{k+1} = \mathbf{M}\bar{\mathbf{X}}_{k+1}$	$nq(2m+1)$	nq	
	$\mathbf{M}_{k+1} = \bar{\mathbf{X}}_{k+1}^{\mathrm{T}}\bar{\mathbf{Y}}_{k+1}$	$\frac{1}{2}nq(q+1)$	$\frac{1}{2}nq(q+1)$	Algorithm is implemented as out-of-core solver
	$\mathbf{K}_{k+1}\mathbf{Q}_{k+1} = \mathbf{M}_{k+1}\mathbf{Q}_{k+1}\boldsymbol{\Omega}_{k+1}^2$	$O(q^3)$ neglected		
	$\mathbf{Y}_{k+1} = \bar{\mathbf{Y}}_{k+1}\mathbf{Q}_{k+1}$	nq^2	nq^2	
Sturm sequence check	$\bar{\mathbf{K}} = \mathbf{K} - \mu\mathbf{M}$	$n(m+1)$	n	
	$\bar{\mathbf{K}} = \mathbf{LDL}^{\mathrm{T}}$	$\frac{1}{2}nm^2 + \frac{3}{2}nm$	$\frac{1}{2}nm^2 + \frac{3}{2}nm$	
Total for solution of p lowest eigenvalues and associated eigenvectors assuming that eight iterations are required and $q = \min\{2p, p+8\}$		$nm^2 + 4nm + 16nq(2m+q+\frac{3}{2})$	$nm^2 + 3nm + 16nq(m+q+\frac{3}{2})$	

The total number of iterations required depends on how 'close' the starting subspace is to the p-dimensional least dominant subspace of the operators and, of course, on the required accuracy of the eigenvalues and associated eigenvectors. Also, it should be noted that in exact arithmetic convergence to an eigenvector is not possible if the starting vectors are all orthogonal to the eigenvector. It is therefore most important to establish a 'good' starting subspace. But there is no need to find for the columns in $\mathbf{X}_1$ vectors each of which is 'close' to a required eigenvector. In the implementation a scheme has proven very effective which, for the special case when $\mathbf{K}$ and $\mathbf{M}$ are diagonal, establishes a starting subspace, which is the least dominant one of the operators. At convergence error bounds on the eigenvalues can be evaluated and a Sturm sequence check can be applied to verify the results. As the solution accuracy for the lowest eigenvalues and corresponding vectors is highest, in general, a four to five digit accuracy in the pth eigenvalue can be sufficient.

Table IV summarizes the algorithm and gives the number of operations required for solution. Based on the experience with the algorithm it is assumed that eight iterations are required. Also, it is assumed that the number of required eigenvalues and vectors is much smaller than the order of the matrices. In this case the solution for the eigensystem of the subspace operators requires a negligible amount of operations.

The subspace iteration solution is most efficient in the analysis of systems with large bandwidth and in out-of-core solutions because relatively little tape handling is necessary. For the routine developed the high speed storage requirements are small, and the lowest eigenvalues and corresponding vectors of very large systems can be calculated. However, it should be noted that the actual cost of an out-of-core solution also includes the cost of the Peripheral Processor (tape and disc reading) time. This time is very system and programming dependent and is not mentioned in Table IV.

SELECTION OF SOLUTION TECHNIQUE

The appropriate solution technique for a given problem should be selected by considering the information given in Tables I–IV. The choice for a solution routine is governed by the number of operations needed for solution and the required high speed storage. The Householder–QR–inverse iteration solution, the generalized Jacobi iteration and the determinant search method have been presented as in-core solution routines because they are likely to be used on systems which can be solved in the high speed storage of the computer. If the

techniques are implemented in out-of-core solution routines, relatively much tape handling is necessary, but the high speed storage requirements would be small.

The generalized Jacobi iteration is most efficient when the complete eigensystem is required and either not many off-diagonal elements are present or they are already small, i.e. the eigenproblem is already 'nearly' solved. For this reason the technique is efficiently used for the solution of the eigensystem of $\mathbf{K}_{k+1}$ and $\mathbf{M}_{k+1}$ in the subspace iteration, equation (34). When the order of the matrices is relatively small, the solution of the eigenvalue problem is not very expensive and the Jacobi iteration may also be attractive because of its simplicity and elegance of solution.

The Householder–QR–inverse iteration solution is most efficient when all eigenvalues and eigenvectors of a matrix are required which has a large bandwidth or is full. As was pointed out, this solution requirement can arise after static condensation of the massless degrees of freedom. A full matrix is also obtained if the generalized eigenvalue problem with a banded mass matrix is transformed to the standard form.

Whether the procedure of static condensation and solution of the reduced eigenvalue problem is efficient depends on the original bandwidth of the stiffness matrix, the increase in bandwidth due to static condensation, the number of original and final degrees of freedom and the number of required eigenvalues and vectors. In most analyses mass is associated with about one half or more of the degrees of freedom; therefore, if the order of the system is large, the static condensation still leads to a large order system which may have lost the bandform. In this case a direct solution of the eigenvalue problem which takes full advantage of the banding characteristics and solves only for the required eigenvalues and associated vectors is more efficient.

When the mass matrix is banded and the system is large the transformation to the standard eigenvalue problem is practically always very inefficient.

The determinant search technique is very effectively used to calculate the lowest eigenvalues and corresponding vectors of systems with small bandwidth. In the solution the eigenvalues and vectors are calculated to high precision. If compacted storage is used also relatively large order systems can be solved in core. The use of a banded mass matrix increases the cost of solution relatively little. Note also that depending on the bandwidth to find the complete eigensystem the determinant search method can be more efficient than the Householder–QR–inverse iteration solution.

The subspace iteration solution is very efficient in the calculation of the lowest eigenvalues and corresponding eigenvectors of systems with large bandwidth and which are too large for the high speed storage of the computer. Note, however, that the eigensystem of the projected operators in equation (34) is calculated in high speed storage, and that, in case many vectors are calculated, this high speed storage requirement may solution govern the problem size.

The most important eigenvalue problem in dynamic analysis is the solution of the lowest eigenvalues and corresponding eigenvectors. However, in some dynamic analyses eigenvalues within a specified intermediate range only are of interest.[17] If the order of the matrices is not large, a solution using the Householder–QR–inverse iteration technique or the determinant search method is efficient, unless only a few eigenvalues and corresponding eigenvectors are needed. In that case a bisection technique such as described in References 11 and 16 can be economical. When the order and bandwidth of the matrices is large, a subspace iteration solution with a shift should be carried out.[2]

The above considerations for the choice of the appropriate algorithm are also applicable to buckling analysis, equation (4). As only one eigenvalue is required, for large order systems the Householder–QR inverse iteration solution and the Jacobi method are obviously inefficient. Depending on the order and bandwidth of the system either the determinant search or the subspace iteration method provides the more efficient solution.

SAMPLE SOLUTIONS

The sample solutions summarized in Table V are actual practical analyses. They have been selected to show typical solution times. In the Jacobi, Householder–QR–inverse iteration and the determinant search solution the eigenvalues have been obtained to near full word precision (12 digits). The subspace iteration

Table V. Sample solutions

Sample number	System	System order n	Maximum half bandwidth m	Mass matrix	Number of required vectors	Solution technique	Computer used	Central processor seconds
1	Plate-beam system	50	Full	Full	50	Jacobi	CDC 7600	2
2	General system	100	Full	Banded positive definite	100	Householder–QR–inverse iteration	CDC 6400	46
3	Plane frame	297	29	Diagonal semi-definite	3	Determinant search	CDC 6400	40
4	Plane frame	297	29	Diagonal semi-definite	3	Subspace iteration	CDC 6400	25
5	Three-dimensional building frame	468	155	Diagonal semi-definite	4	Subspace iteration	CDC 6400	160
6	Three-dimensional box	403	114	Diagonal semi-definite	8	Subspace iteration	CDC 7600	12
7	Dam	724	113	Banded	20	Subspace iteration	CDC 6600	518
8	Dam	226	68	Banded	7	Determinant search	CDC 6600	71
9	Piping system	566	11	Diagonal semi-definite	28	Subspace iteration	CDC 6600	142
10	Piping system	566	11	Diagonal semi-definite	7	Determinant search	CDC 6600	11
11	Building with foundation	1174	137	Diagonal semi-definite	45	Subspace iteration	CDC 6600	890
12	Building	340	31	Diagonal semi-definite	7	Determinant search	CDC 6600	20
13	Bridge	342	35	Diagonal semi-definite	10	Subspace iteration	CDC 6600	31

solved in each case for the largest required eigenvalue to about 5 digit precision with the lower eigenvalues being more accurate. Note that the central processor speeds of the CDC 6400, CDC 6600 and CDC 7600 computers are approximately as 1 : 3 : 8.

The Jacobi iteration was only used on rather small order systems such as indicated in the table. The size of the systems which have been solved by the Householder–QR–inverse iteration and the determinant search technique was restricted by the maximum high speed storage available. As indicated in Tables I and III the determinant search method can solve larger order systems. The subspace iteration solution has been used in most cases because the algorithm has been programmed to allow practically unlimited system size and bandwidth.

The solution times in Table V can only be used as a guide to estimate the computer effort involved in using the appropriate algorithm in a required analysis. The table does not demonstrate the relative efficiencies of the different solution techniques when used on the same problems. Tables I–IV do this and it would be too expensive to run comparative example analyses merely to arrive at the same conclusions. However, the solution times do emphasize the points made in the previous section about the selection of the appropriate algorithm for a given problem.

CONCLUSIONS

A single algorithm which always gives a very efficient solution of the generalized eigenvalue problems does not exist. In this paper the probably most efficient solution methods currently in use have been summarized. An efficient solution of a specific eigenvalue problem is obtained if the appropriate one of these methods is used.

The Householder–QR–inverse iteration technique is a general method for the solution of standard eigenvalue problems and requires a transformation into this form.

The generalized Jacobi iteration, the determinant search method and the subspace iteration algorithm have been developed specifically for direct solution of the generalized eigenvalue problems. The methods are very efficient because advantage is taken of the specific solution requirements and the specific properties of the stiffness and mass matrices, e.g. the banding characteristics, the relative magnitude and the relative positioning of the elements in the matrices. Using the specific properties of the matrices it appears that much potential lies in the subspace iteration solution. The starting subspace generated by the algorithm has proven to be very effective, i.e. only about eight iterations are required for convergence. However; the potential of the method lies in that a 'better' starting subspace would further reduce the number of required iterations, and it is felt that future research should be directed towards this aim.

REFERENCES

1. J. S. Archer, 'Consistent mass matrix for distributed mass systems', *Proc. ASCE*, **89**, *J. Str. Div.* **4**, 161–178 (1963).
2. *K. J.* Bathe, 'Solution methods for large generalized eigenvalue problems in structural engineering', *SESM Rep.*, 71–20, Civ. Engng Dept., Univ. of Calif., Berkeley, U.S.A. (1971).
3. K. J. Bathe and E. L. Wilson, 'Eigensolution of large structural systems with small bandwidth', *ASCE, J. Engng. Mech. Div.* (to appear).
4. K. J. Bathe and E. L. Wilson, 'Large eigenvalue problems in dynamic analysis', *ASCE, J. Engng. Mech. Div.* **98**, 1471–1485 (1972).
5. F. L. Bauer, 'Das Verfahren der Treppen-Iteration und Verwandte Verfahren zur Lösung Algebraischer Eigenwertprobleme', *ZAMP* **8**, 214–235 (1957).
6. O. E. Brönlund, 'Eigenvalues of large matrices', Symp. *Finite Element Techniques* at Inst. für Statik und Dynamik der Luft und Raumfahrtskonstruktionen, Univ. of Stuttgart, Germany (1969).
7. R. W. Clough, 'Analysis of structural vibrations and dynamic response', Proc. U.S.–Japan Symp. *Recent Advances in Matrix Methods of Structural Analysis and Design*, Tokyo, Japan (1969).
8. S. B. Dong, J. A. Wolf and F. E. Peterson, 'On a direct-iterative eigensolution technique', *Int. J. num. Meth. Engng.* **4**, 155–162 (1972).
9. S. Falk and P. Langemeyer, 'Das Jacobische Rotationsverfahren für Reell-symmetrische Matrizenpaare', *Elektron. Datenverarb.* 30–34 (1960).
10. C. A. Felippa, 'BANEIG-eigenvalue routine for symmetric band matrices', *Comp. Prog. Ser., SESM*, Civ. Engng Dept., Univ. of Calif., Berkeley, U.S.A. (1966).
11. K. K. Gupta, 'Vibration of frames and other structures with banded stiffness matrix', *Int. J. num. Meth. Engng.* **2**, 221–228 (1970).

12. A. Jennings, 'A direct iteration method of obtaining latent roots and vectors of a symmetric matrix', *Proc. Camb. Phil. Soc.* **63**, 755–765 (1967).
13. A. Jennings and D. R. L. Orr, 'Application of the simultaneous iteration method to undamped vibration problems', *Int. J. Num. Meth. Engng.* **3**, 13–24 (1971).
14. J. M. Ortega and H. F. Kaiser, 'The LL^T and QR methods for symmetrictri diagonalm atrices', *Comp. J.* **6**, 99–101 (1963).
15. B. N. Parlett and W. Kahan, 'On the convergence of a practical QR algorithm', Proc. IFIP Congress (1968).
16. G. Peters and J. H. Wilkinson, '$Ax = \lambda Bx$ and the generalized eigenproblem', *SIAM, J. Num. Anal.* **7**, 479–492 (1970).
17. M. F. Rubinstein, *Structural Systems—Statics, Dynamics and Stability*, Prentice-Hall, Englewood Cliffs, New Jersey, 1970.
18. H. Rutishauser, 'Computational aspects of F. L. Bauer's simultaneous iteration method', *Num. Math.* **13**, 4–13 (1969).
19. J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.
20. O. C. Zienkiewicz, *The Finite Element Method in Engineering Science*, McGraw-Hill, London, 1971.