ACCELERATION OF CONVERGENCE IN SOLVING THE EIGENVALUE PROBLEM BY MATRIX ITERATION USING THE POWER METHOD

by

Julio Cesar Massa

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
in
Engineering Mechanics

APPROVED:

L. Meirovitch, Chairman

M. P. Singh

D. T. Mook

October, 1985

Blacksburg, Virginia

ACCELERATION OF CONVERGENCE IN SOLVING THE EIGENVALUE PROBLEM BY MATRIX ITERATION USING THE POWER METHOD

by

Julio C. Massa

(abstract)

A modification of the matrix iteration using the power method, in conjunction with Hotelling deflation, for the solution of the problem $K.x = \omega^2.M.x$ is here propose. The problem can be written in the form $D.x = \lambda.x$, and the modification consists of raising the matrix D to an appropriate power p before carrying out the iteration process.

The selection of a satisfactory value of p is investigated, based on the spacing between the eigenvalues. The effect of p on the accuracy of the results is also discussed.

ACKNOWLEDGEMENTS

I would like to thank Dr. Leonard Meirovitch, University Distinguished Professor in the ESM Department, Virginia Tech, for suggesting the topic for this thesis and giving able guidance for carrying out the research. I also appreciate the time that Dr. M. P. Singh and Dr. D. T. Mook gave me in serving on my thesis committee.

I am grateful to the Argentine Universities of Cordoba and Rio Cuarto for providing partial support for my higher education.

Finally, I express my gratitude to Dr. Carlos A. Prato, who introduced me to field of structural vibrations.

CONTENTS

1. INTRODUCTION

In determining the response of a structure by modal analysis, it is necessary to know the system modes. This requires the solution of the eigenvalue problem

$$Kx = \omega^2 Mx \tag{1.1}$$

where K and M are real and symmetric stiffness and mass matrices, respectively.

There exists a great variety of methods for solving (1.1). The efficiency of these methods depends mostly on the order of the system and the properties of the matrices K and M.

The solution of the eigenvalue problem is computationally expensive, so that it is desirable to develop more efficient techniques. Our goal is to improve the matrix iteration method. Before, we discuss the nature of the improvement, we propose to review matrix iteration briefly.

1.1 MATRIX ITERATION BY THE POWER METHOD

Assuming that the stiffness matrix is nonsingular, we can rewrite (1.1) as

$$Dx = \lambda x \tag{1.2}$$

where D is a real nonsymmetric matrix

$$D = K^{-1}M \tag{1.3}$$

and

$$\lambda = 1/\omega^2 \tag{1.4}$$

is a real parameter.

Next, we define the iterative scheme

$$v_s = D\, v_{s-1}\,, \quad s = 1,2,\ldots \tag{1.5}$$

It can be shown (Ref. 1) that in the limit, the iterated vector v_s converges to the first eigenvector x_1 belonging to the dominant eigenvalue λ_1, or

$$\lim_{s\to\infty} v_s = x_1 \tag{1.6}$$

where the first eigenvalue λ_1 can be obtained from

$$v_{s+1} = \lambda_1 v_s \tag{1.7}$$

so that, after achieving convergence, the dominant eigenvalue can be computed as the ratio of homologous elements of two consecutive iterated vectors.

It is important to notice that this process is not suitable for determining eigenvalues other than the dominant one. However, the process can be modified to obtain the remaining eigenvalues and the corresponding eigenvectors.

1.2 HOTELLING'S DEFLATION

We shall describe here a technique due to Hotelling, known as matrix deflation.

The eigenvectors x_i ($i = 1,2,\ldots,n$) are orthogonal and can be normalized such that

$$x_i^T M x_j = \delta_{ij} \quad i,j = 1,2,\ldots,n \tag{1.8}$$

Next, we define the deflated matrix

$$D_2 = D - \lambda_1 x_1 x_1^T M \tag{1.9}$$

It can be shown (Ref. 1) that using the matrix D_2 in the iterative scheme (1.5), leads to the second dominant eigenvalue λ_2 and simultaneously to the second eigenvector.

This procedure can be generalized to compute as many eigenvalues as desired, by defining the deflated matrix (Ref. 1).

$$D_k = D_{k-1} - \lambda_{k-1} x_{k-1} x_{k-1}^T M, \quad k = 2,3,\ldots,m \tag{1.10}$$

which yields the eigenvalue λ_k and the corresponding eigenvector x_k.

It is important to notice that the deflated matrix D_k is based on the previously computed eigenvectors $x_1, x_2, \ldots, x_{k-1}$, so that accuracy decreases with each new eigenvalue, eigenvector pair. This aspect represents the main drawback of Hotelling deflation, and, as discussed

in the following chapter, it places a serious limitation upon the value of p, and consequently upon the efficiency of the proposed modification.

1.3 PROPOSED MODIFICATION

Let us go back to the iterative scheme defined by (1.5). After p steps, the iterated vector is

$$\begin{aligned} v_p &= Dv_{p-1} \\ &= D(Dv_{p-2}) \\ &= D[D(Dv_{p-3})] \\ &\quad \vdots \\ &= D^p v_0 \end{aligned} \tag{1.11}$$

where v_0 is an arbitrary initial vector, which is often defined as having all its components equal to one.

Next, we propose to replace (1.5) by the following iterative sequence

$$v_k = D^p v_{k-1} \tag{1.12}$$

It is obvious that each step in (1.12) is exactly equivalent to p steps using (1.5). However, the extra cost is relatively high because we must

raise D to the power p, and we only save (p-1) multiplications of a matrix times a vector in each step. Because the main objective of this work is to economize computational effort, we must count the number of multiplications involved in each method, to reveal advantages or disadvantages.

Next, let m be the number of steps required to reach convergence using (1.5). Noticing that a matrix multiplication requires n^3 multiplications, while a multiplication of a vector by a matrix requires only n^2 multiplications, the number of operations in each case being

$$N_s = m\, n^2 \tag{1.13}$$

using (1.5) and

$$N_p = (p-1)n^3 + mn^2/p \tag{1.14}$$

using (1.12). For the modification to be effective, it is required that $N_p < N_s$, and hence,

$$m > pn \tag{1.15}$$

For accelerated convergence in the case p = 2, it is necessary for m to be greater than 2n. Because this is not always the case, at this stage the new method does not appear attractive.

The situation looks much more promising if an iteration of the type (1.12) can be used for the subdominant eigenvalues. This implies using D^p in defining the deflated matrices. In this regard, we notice that the eigenvalues of D^p are related to the eigenvalues of D as

$$D^p x_i = (\lambda_i)^p \, xi \tag{1.16}$$

while the eigenvectors are the same. Hence, the deflated matrix for the second eigenvalue can be redefined as

$$D_{2p} = D^p - \lambda_{1p} \, x_1 x_1^T \, M \tag{1.17}$$

where

$$\lambda_{1p} = (\lambda_1)^p \tag{1.18}$$

At this point, it is important to recognize that the "new deflated matrix" D_{2p} is not equal to D_2, as defined in (1.9), raised to the power p. Indeed,

$$D_{2p} \neq (D_2)^p \tag{1.19a}$$

because

$$D^p - (\lambda_1)^p \, x_1 x_1^T \, M \neq (D - \lambda_1 x_1 x_1^T \, M)^p \tag{1.19b}$$

If we were to use the deflated matrix defined by the right hand side of (1.19), we could anticipate that the numerical results would be identical to those obtained by the procedure based on (1.10). However it would be necessary to raise the deflated matrix to the power p in each step, and hence the cost would be very high, as discussed earlier.

When using the modified deflated matrix defined by (1.17), we have the advantage that the matrix D is raised to the power p only once, and we can economize on steps during the search for all the eigenvalues. This is so, because (1.17) can be generalized to obtain the remaining eigenvalues by defining the modified deflated matrix.

$$D_{kp} = D_{(k-1)p} - \lambda_{(k-1)p} x_{k-1} x_{k-1}^T M , \qquad k = 2,3,\ldots,m \qquad (1.20)$$

Although inequality (1.19) holds true for the deflated matrix defined by (1.20), it is important to recognize that we would obtain the same results as obtained by the standard procedure based on (1.10). This is so, because using the modified process is equivalent to solving the modified problem (1.16) by the standard process.

The fact that we can obtain only approximate numerical solutions, causes us to expect discrepancies between the results of the two processes, due to inequality (1.19). This aspect will be discussed later.

Next, we go back to the task of counting operations to find out if we can expect computational economy by using the proposed moficiation. To this end, let $\bar{m}$ be the average number of iterations required by the standard procedure to achieve convergence for a given tolerance and r be the desired number of eigenvalues to be determined.

Equations (1.13) and (1.14) for the total number of multiplication become

$$N_s = r\,\bar{m}\,n^2 \qquad (1.21)$$

using (1.10) and

$$N_p = (p-1)n^3 + \frac{r\bar{m}\,n^2}{p} \qquad (1.22)$$

using (1.20). After some algebraic manipulations, we can write the efficiency in computational effort, E, as

$$E = \frac{N_s - N_p}{N_s} = 1 - \left[\frac{(p-1)n}{r\,\bar{m}} + \frac{1}{p}\right] \qquad (1.23)$$

Defining the parameter

$$R = \frac{r\,\bar{m}}{n} \tag{1.24}$$

we can rewrite (1.23) as

$$E = 1 - \frac{1}{p} - \frac{(p-1)}{R} \tag{1.25}$$

In cases where $r.\bar{m}$ is relatively large relative to $(p-1).n$, equation (1.23) simplifies to

$$E \approx 1 - \frac{1}{p} \tag{1.24}$$

This result could actually be anticipated. It reiterates the fact that the extra work done to obtain D^p becomes negligible when we must compute a large number of eigenvalues, and each one requires a large number of iterations. In such cases, due to the modification, the work reduces from 1 to 1/p.

It is important to note that in cases where $r.\bar{m}$ is not large compared to $(p-1).n$, we can have a negative value for the efficiency E, meaning that the proposed method is less efficient than the standard method.

Equation (1.25) can be plotted for different values of p, obtaining a graph of the type shown in FIGURE 1. For a given problem there exist a value P_1 for p which is optimum and a value P_2 beyond which the modification becomes completely inefficient.

Using different values of the problem parameter R, we can plot graphs like the one shown in FIGURE 2, where we see that the efficiency

increases when R increases, and hence the modification is more efficient. It is important that, as R increases, a convenient value of p can be chosen within a relative large interval.

At this stage, the modified process looks attractive enough to warrant further investigation. The next step will be to investigate how the value of p affects the accuracy of the results.

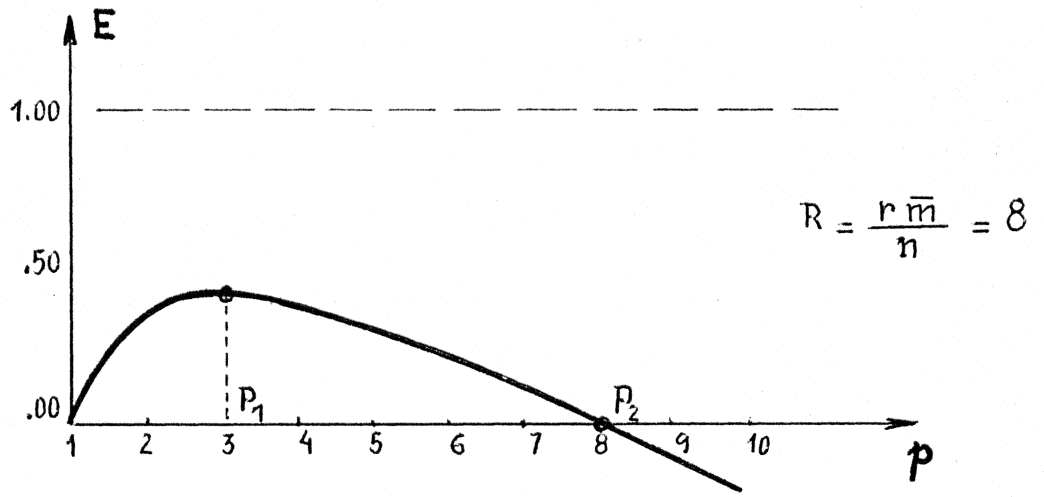

FIGURE 1 COMPUTATIONAL EFFICIENCY AS A FUNCTION OF p

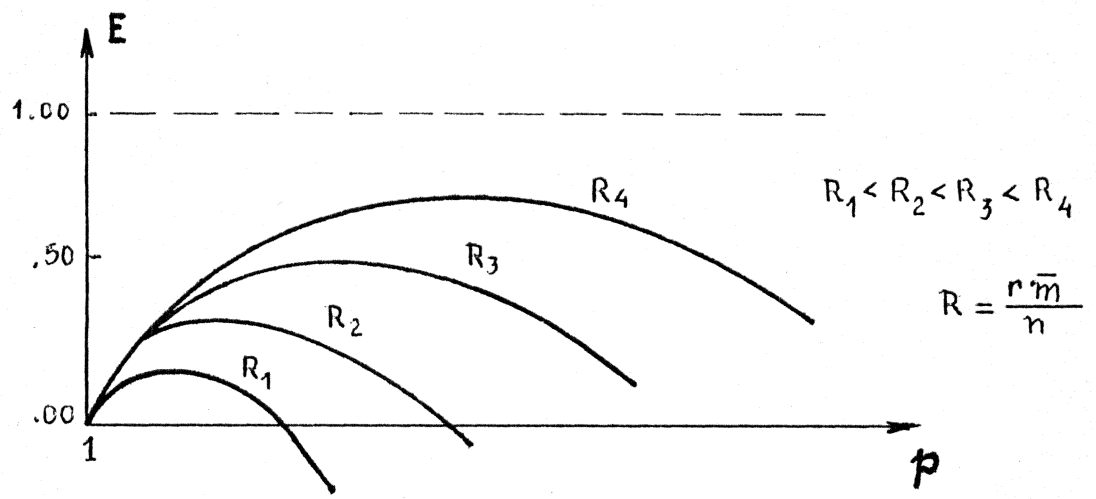

FIGURE 2 COMPUTATIONAL EFFICIENCY FOR DIFFERENT VALUES OF R

2. ERROR ANALYSIS

Our objective is to investigate how the value of p used in (1.12) affects the accuracy of the solution of the eigenvalue problem (1.2).

2.1 ERROR IN THE COMPUTED FIRST EIGENVECTOR

We use lower case letters to denote computed values and capital letters to denote exact values. Let ε_{k+1} be the coordinate of the computed k eigenvector on the k+1 exact eigenvector.

The computed first eigenvector is

$$x_1 = X_1 + \varepsilon_2 X_2 + \alpha_3 X_3 + \ldots \tag{2.1}$$

where ε_2 is related to the prescribed tolerance used to control convergence in the eigenvector. Provided that Λ_3/Λ_2 is not much smaller than Λ_2/Λ_1, we can assume that α_3 is of order ε^2. Hence, keeping only first-order terms in ε

$$x_1 = X_1 + \varepsilon_2 X_2 \tag{2.2}$$

We can compute the eigenvalue using Rayleigh's quotient. Recalling that the eigenvectors have been normalized according to (1.8),

$$\lambda_1 = x_1^T K x_1$$

$$= (X_1 + \varepsilon_2 X_2)^T K (X_1 + \varepsilon_2 X_2)$$

$$= X_1^T K X_1 + \varepsilon_2 X_1^T K X_2 + \varepsilon_2 X_2^T K X_1 + o(\varepsilon_2^2) \qquad (2.3)$$

Invoking the orthogonality property and keeping first-order terms in ε,

$$\lambda_1 = \Lambda_1 \qquad (2.4)$$

Equations (2.2) and (2.4) are first-order expansions for the first computed eigenvector and eigenvalue.

2.2 ERROR IN THE COMPUTER SECOND EIGENVECTOR

To determine how the error in the first computed eigenvector propagates to the second, we must first expand the computed deflated matrix $\hat{D}_{2p}$ in a series including only first-order terms in ε, or

$$\hat{D}_{2p} = D^p - (\lambda_1)^p [x_1 x_1^T M]$$

$$= D^p - \Lambda_1^p [X_1 X_1^T + \varepsilon_2 X_2 X_1^T + \varepsilon_2 X_1 X_2^T] M \qquad (2.5)$$

Invoking the expansion theorem (Ref. 1), we can write the k iterated vector in the form

$$\bar{v}_k = a_{1k} X_1 + X_2 + a_{3k} X_3 + a_{4k} X_4 + \ldots \qquad (2.6)$$

which implies a certain normalization process. The next iterated vector can be determined using (2.5) and (2.6). Upon using the orthogonality property, and keeping first-order terms in ε,

$$v_{k+1} = \hat{D}_{2p}\bar{v}_k$$

$$= (-\varepsilon_2\Lambda_1^p)\, X_1 + (\Lambda_2^p - \varepsilon_2\Lambda_1^p)\, X_2 + \sum_{j=3}^{n} a_{jk}\Lambda_j^p\, X_j \qquad (2.7)$$

Dividing (2.7) by Λ_2^p and normalizing the iterated vector as in (2.6), we obtain

$$\bar{v}_{k+1} = \left(\frac{-e_1}{1 - e_1}\right) X_1 + X_2 + \sum_{j=3}^{n} \frac{a_{jk}}{1 - e_1} \left(\frac{\Lambda_j}{\Lambda_2}\right)^p X_j \qquad (2.8)$$

where e_1 is a constant having the value

$$e_1 = \varepsilon_2 \left(\frac{\Lambda_1}{\Lambda_2}\right)^p \qquad (2.9)$$

CASE $e_1 << 1$. This is the case when p is small. Eqation (2.8) becomes

$$\bar{v}_{k+1} = -e_1\, X_1 + X_2 + \sum_{j=3}^{n} a_{jk} \left(\frac{\Lambda_j}{\Lambda_2}\right)^p X_j \qquad (2.10)$$

The coordinates in the higher eigenvectors become smaller with each iteration, because $(\Lambda_j/\Lambda_2)^p << 1$. However, the coordinate in the first eigenvalue remains constant during the whole process.

In this case, convergence to the second computed eigenvalue yields

$$x_2 = -e_1 X_1 + X_2 + \varepsilon_3 X_3 + \alpha_4 X_4 \qquad (2.11)$$

Here we can repeat the same considerations as for the first computed eigenvector. We assume that α_4 is of order ε^2 while ε_3 is controlled by the tolerance.

Case $e_1 >> 1$. This is the case when Λ_1 and Λ_2 are well separated and p is very large. The coordinate in the first eigenvector remains constant and equal to one. Hence, the result is completely unacceptable.

Case $e_1 \approx 1$. In this case the coordinate in the first eigenvector is very large and remains constant. Hence, the result is again unsatisfactory. Depending on what happens with the coordinate in the third eigenvector, the process could converge to the first eigenvalue, eigenvector pair.

The above discussion shows that the method fails if e_1 is not small compared to one. This fact rules out large values of p.

For the case $e_1 << 1$, we can use Rayleigh's quotient and compute the second eigenvalue as follows:

$$x_2 = - e_1 X_1 + X_2 + \varepsilon_3 X_3 \tag{2.12}$$

$$\therefore \quad \lambda_2 = x_2^T K x_2$$

$$= (e_1)^2 \Lambda_1 + \Lambda_2 + (\varepsilon_3)^2 \Lambda_3 \tag{2.13}$$

Assuming that e_1 is of order ε and keeping only first-order terms in ε,

$$\lambda_2 = \Lambda_2 \tag{2.14}$$

2.3 ERROR IN THE COMPUTED THIRD EIGENVECTOR

Assuming that the method behaved well when computing the second eigenvalue, that is $e_1 << 1$, we compute the deflated matrix for the third eigenvector

$$\hat{D}_{3p} = \hat{D}_{2p} - \Lambda_2^p [x_2 x_2^T] M \tag{2.15}$$

The normalized iterated vector can be written as,

$$\bar{v}_k = a_{1k} X_1 + a_{2k} X_2 + X_3 + a_{4k} X_4 + \ldots \tag{2.16}$$

Upon invoking the orthogonality conditions, keeping only first-order terms in ε and recalling (2.9), we can write the subsequent normalized iterated vector as

$$\bar{v}_{k+1} = 0\, X_1 + \left(\frac{-e_2}{1 - e_2 a_{2k}}\right) X_2 + X_3 + \sum_{j=4}^{n} \frac{a_{jk}}{1 - e_2 a_{2k}} \left(\frac{\Lambda_j}{\Lambda_3}\right)^p X_j \tag{2.17}$$

where the constant e_2 has the value

$$e_2 = \varepsilon_3 \left(\frac{\Lambda_2}{\Lambda_3}\right)^p \tag{2.18}$$

To obtain convergence in the computed third eigenvector, the coordinate a_{2k} must converge. Hence, for $k \to \infty$

$$\frac{-e_2}{1 - e_2 \cdot a_{2\infty}} = a_{2\infty} \tag{2.19}$$

Assuming that $(e_2 \cdot a_{2\infty}) << 1$

$$a_{2\infty} = - e_2 \tag{2.20}$$

CASE $e_2 << 1$. Convergence will be reached and the computed third eigenvalue leads to

$$x_3 = - e_2 X_2 + X_3 + \varepsilon_4 X_4 \tag{2.21}$$

CASES $e_2 \approx 1$ and $e_2 >> 1$. Equation (2.19) show that the computed third eigenvalue will be completely unsatisfactory.

2.4 ERROR IN THE REMAINING EIGENVECTORS

A similar analysis, when carried out for the remaining computed eigenvectors, shows that, if $e_s << 1$, the method behaves well and the computed eigenvectors are,

$$\begin{aligned} x_4 &= - e_3 X_3 + X_4 + \varepsilon_5 X_5 \\ x_5 &= - e_4 X_4 + X_5 + \varepsilon_6 X_6 \\ &\vdots \\ x_s &= - e_{s-1} X_{s-1} + X_s + \varepsilon_{s+1} X_{s+1} \end{aligned} \tag{2.22}$$

where e_{s+1} is controlled by the tolerance and

$$e_{s+1} = \varepsilon_s \left(\frac{\Lambda_{s-1}}{\Lambda_s}\right)^p << 1 \tag{2.23}$$

Equations (2.22) and (2.23) are of greatest importance in this error analysis. These results are valid for the modified process as well as for the standard process, p=1.

Equation (2.23) shows that the error grows with the power p. Hence, the accuracy suffers. Recalling that ε_{s+1} is controlled by the tolerance, we are left with two causes of error increase; one is the power p and the other is the ratio of two neighboring eigenvalues.

CONCLUSIONS

- When the method behaves well, the error is given by equation (2.23). This behavior takes place for small values of p.

- For very large values of p, the method will definitely fail.

- The behavior of the method for intermediate values of p cannot be predicted by this first-order analysis.

3. TESTING THE METHOD

3.1 COMPUTER PROGRAM

The computer program was based on the equations developed in Chapter 1. It will be described only briefly, because the power method and Hotelling deflation are well known techniques (Ref. 1). Moreover, the modification consists only of raising the matrix D to a certain power p.

Assuming that the matrix D is given as input, the program consists of the following steps:

Loop to raise the matrix D to the power p.

Loop to obtain the desired number of eigenvectors.

1. Definition of the initial vector v_0.
2. Iteration loop to achieve convergence
   - * Computation of the iterated vector.
   - * Normalization of the iterated vector.
   - * Control of convergence.
3. Orthonormalization of the eigenvector.
4. Computation of the eigenvalue.
5. Computation of the deflated matrix.

Having described the general configuration of the program, we can explain some of the details.

To save computational effort, D is first squared, then the square of the square is computed, etc. In doing so, we obtain successively D^2, D^4, D^8, D^{16},... . This approach is very efficient when

p is large, although it places a limitation on the possible values of p. The initial vector is defined as having all its components equal to one. The iterated vector is normalized such that the greatest component is equal to one. Convergence is achieved when for every component the absolute value of the difference between homologous components of two consecutive iterated vectors is less than a given tolerance.

Recalling (1.4) and (1.18), the frequencies are computed according to

$$\omega_i = (\lambda_{ip})^{-1/2p} \tag{3.1}$$

Care must be taken to avoid overflow when using large values of p. Each time the matrix is squared, it is also normalized such that the greatest component on the diagonal is made equal to one. In this case the eigenvalues are multiplied by the normalizing factor.

3.2 NUMERICAL EXAMPLE 1. VIBRATION OF A MEMBRANE

To carry out the first test of the method, it is convenient to solve a problem for which the matrix D could be easily computed. It is also convenient that the required number of iterations be large, so that the effect of using different values for p could be better appreciated. The problem chosen consisted of the eigenvalue problem for a rectangular membrane.

3.2.1 DEFINITION OF THE PROBLEM

A rectangular membrane clamped at the boundaries x=0, x=1, y=0, y=1/2, is subject to a uniform tension T=1 and has mass per unit area

$$\rho(x,y) = (1 + x^2)(1 + 4y^2) \tag{3.2}$$

We propose to solve the eigenvalue problem by Rayleigh-Ritz method using 25 admissible functions of the form

$$\phi_i = \sin m\pi x \cdot \sin n\pi 2y , \qquad m,n = 1,2,\ldots,5$$

$$\phi_j = \sin p\pi x \cdot \sin q\pi 2y , \qquad p,q = 1,2,\ldots 5 \tag{3.3}$$

The entries of the stiffness matrix are

$$K_{ii} = \frac{1}{8}\pi^2(p^2 + 4q^2)$$

$$K_{ij} = 0 , \text{ for } i \neq j \tag{3.4}$$

and the entries of the mass matrix are

$$M_{ij} = \frac{1}{2} A_{nq} \cdot A_{mp} \tag{3.5}$$

where

$$A_{rs} = \begin{cases} I_{rs} + \frac{1}{2} & \text{for } r = s \\ I_{rs} & \text{for } r \neq s \end{cases} \tag{3.6}$$

in which

$$I_{rs} = \int_0^1 z^2 \sin(r\pi z) \cdot \sin(q\pi z)dz , \qquad r,s,=1,2,\ldots 5 \tag{3.7}$$

Next, we can formulate the eigenvalue problem as in (1.1)

$$Kx = \omega^2 Mx \tag{3.8}$$

Finally, noticing that K is diagonal the problem can be written in the standard form (1.2), or

$$Dx = \lambda x, \qquad D_{ij} = M_{ij}/K_{ii} \tag{3.9}$$

3.2.2 RESULTS

Table 1 shows the computed eigenvalues and natural frequencies and, as expected, some of them are very close. In column four we have the ratio between adjacent eigenvalues. The lowest ratio is R=1.003 corresponding to the 19^{th} eigenvalue, which is consistent with the fact that the maximum number of iterations occurs when computing the 19^{th} eigenvalue, eigenvector pair.

Table 2 shows the number of iterations required to achieve convergence for different values for p. Tolerance was set equal to $.2 \times 10^{-5}$ for every component of the normalized eigenvector. We see that the number of iterations reduces to one half when using D^2, to one fourth when using D^4, etc. For very large values of p, the process is unable even to produce the second eigenvector. Recalling (2.9), this can be explained by the fact that e_1 is not small compared to one. For example, when p=64, $(\Lambda_1/\Lambda_2)^p = 3.5 \times 10^{13}$.

For intermediate values of p, the method behaves well until it fails at some point, by converging to th first eigenvector or not

converging at all. It was also observed that, when the method converges, the results are fairly accurate.

To obtain higher accuracy, the way of computing the eigenvalues was changed. The use of Rayleigh's quotient produced a small improvement, but the main problem did not disappear.

The effect of changing the tolerance was also investigated. Table 3 shows the effect the tolerance has over the first failure of the process for increasing values of p. In several cases, the first failure occurred when computing the 11^{th} eigenvector. This behavior can be explained by the fact that the preceding eigenvalues (9^{th} and 10^{th}) are very close, R=1.02. A similar situation was observed for the 16^{th} eigenvector.

3.2.3 TESTING ERROR PROPAGATION

To verify the validity of equations (2.22) and (2.23), which are the most important result of the error analysis, we began by computing the actual eigenvalues using p=1 and a tolerance of 10^{-13}.

The coefficients of the eigenvector expansion of the computed eigenvectors were obtained invoking the orthogonality conditions

$$x = \sum_{j=1}^{n} a_{sj} X_j \qquad (3.10)$$

$$a_{sj} = X_j^T M x_s \qquad (3.11)$$

We notice that except for j=s all these coefficients correspond to errors. Moreover, using the notation of Chapter 2,

$$e_{s-1} = a_{s(s-1)} \tag{3.12}$$

$$\varepsilon_{s+1} = a_{s(s+1)} \tag{3.13}$$

Table 4 shows the results for the case p=4 and tolerance $= .2 \times 10^{-5}$. The results are in complete agreement with (2.23). To illustrate this fact, we observe that

$$x_{13} = -.00000465\, X_{12} + X_{13} + .00000060\, X_{14}$$

$$x_{14} = -.00000110\, X_{13} + X_{14} + .00004045\, X_{15}$$

Using the eigenvalues from TABLE 1, we can check that

$$e_{13} = -\varepsilon_{14} \left(\frac{\Lambda_{13}}{\Lambda_{14}}\right)^4 = -.00000060 \times \left(\frac{.003712}{.003186}\right)^4 = -.00000110 \tag{3.14}$$

3.3 NUMERICAL EXAMPLE 2. VIBRATION OF A BEAM

To test the method under different circumstances, the problem of a uniform simple supported beam was solved. For this case, the natural frequencies are (Ref. 1)

$$\omega_i = i^2\pi^2 (EI/mL^4)^{1/2}, \qquad i = 1,2,\ldots \tag{3.15}$$

The eigenvalues are very well spaced. Hence, based on (2.23), we can anticipate that the error in this case will be greater than in the case of the membrane.

The mass and stiffness matrices were computed by means of the finite element method using Hermite cubics as interpolation functions. The parameters of the beam used were EI = 10^9, m = 100, L = 1000.

Using 25 elements of equal length, the resulting element matrices are (Ref. 1),

$$K_i = \frac{10^9}{40^3} \begin{bmatrix} 12 & 6 & -12 & 6 \\ & 4 & -6 & 2 \\ \text{symm} & & 12 & -6 \\ & & & 4 \end{bmatrix}$$

$$M_i = \frac{10^5}{420 \times 25} \begin{bmatrix} 156 & 22 & 54 & -13 \\ & 4 & 13 & -3 \\ \text{symm} & & 156 & -22 \\ & & & 4 \end{bmatrix}$$

RESULTS

Table 5 shows that the computed natural frequencies for the model of 50 degree of freedom are in good agreement with the exact solution, especially for the lower frequencies. The last column shows that the ratio between adjacent eigenvalues is large.

Table 6 shows the number of iterations required to achieve convergence, for a tolerance $.2 \times 10^{-8}$, when using different values of

p. The number of iterations, and consequently the computational effort reduces to approximately one half for p=2.

Because the ratio of adjacent eigenvalues is larger than in the case of the membrane, the number of iterations is smaller. Furthermore, according to (2.23) the errors are larger, causing the method to fail for smaller values of p. Using p=3, the process only provides the first 8 eigenvectors and using p=4, only the first 4 eigenvectors.

It is very important to stress the fact that the method failed to converge at the 45^{th} eigenvector when using p=1. This means that the standard method can also fail due to error propagation.

3.4 COMPARISON WITH THE METHOD OF SUBSPACE ITERATION

Two cases were solved by the proposed method and also by the method of subspace iteration (Ref. 3), to compare the time each method used to solve the same problem to approximately the same accuracy.

CASE 1. Computation of the first 15 modes for the beam problem with 50 degree of freedom. Using p=2 for the proposed method, the time ratio was 1 to 3 in favor of subspace iteration.

CASE 2. Computation of the first 7 eigenvectors for the membrane problem with 16 degree of freedom. Using p=8, the proposed method used the same time as the method of subspace iteration. However, it must be clear that in this particular case, inversion of the stiffness matrix was not necessary and the mass matrix was a full matrix.

CONCLUSION: In general the method of subspace iteration is more efficient, because it takes full advantage of the fact that the stiffness and mass matrices are banded and symmetric.

TABLE 1. FREQUENCIES AND EIGENVALUES FOR THE MEMBRANE

p=1 Tolerance = $.2 \times 10^{-5}$

Mode number	frequency ω_i	eigenvalue λ_i	Ratio λ_i/λ_{i+1}
1	5.3613	.034790	1.628
2	6.8401	.021373	1.599
3	8.6486	.013369	1.213
4	9.5260	.011020	1.255
5	10.6710	.008782	1.036
6	10.8637	.008473	1.248
7	12.1367	.006789	1.161
8	13.0749	.005850	1.087
9	13.6334	.005380	1.022
10	13.7797	.005266	1.213
11	15.1788	.004340	1.113
12	16.0165	.003898	1.050
13	16.4127	.003712	1.165
14	17.7161	.003186	1.005
15	17.7626	.003169	1.208
16	19.5267	.002623	1.020
17	19.7248	.002570	1.137
18	21.0328	.002261	1.127
19	22.3304	.002005	1.003
20	22.3663	.001999	1.167
21	24.1619	.001713	1.023
22	24.4330	.001675	1.159
23	26.3072	.001445	1.124
24	27.8960	.001285	1.130
25	29.6485	.001138	--

TABLE 2. NUMBER OF ITERATIONS TO ACHIEVE CONVERGENCE FOR DIFFERENT VALUES OF p

$$\text{tolerance} = .2 \times 10^{-5}$$

Mode number	D	D^2	D^4	D^8	D^{16}	D^{32}	D^{64}
1	27	15	8	5	3	2	2
2	28	15	8	5	3	19	FAILURE
3	57	31	17	9	14	FAILURE	
4	55	29	16	9	FAILURE		
5	287	154	82	44			
6	56	30	16	9			
7	81	44	23	13			
8	114	62	34	18			
9	522	278	148	78			
10	61	33	18	13			
11	118	63	33	FAILURE			
12	214	114	61				
13	76	41	22				
14	1360	746	407				
15	72	38	20				
16	570	303	160				
17	86	46	25				
18	103	55	29				
19	2808	1513	811				
20	87	46	25				
21	335	184	100				
22	103	54	29				
23	108	58	31				
24	109	58	31				
25	3	3	17				

TABLE 3 EFFECT OF THE VALUE OF THE TOLERANCE ON THE FIRST FAILURE FOR INCREASING VALUES OF p

Tolerance in eigenvector components	power p	matrix	First failure at eigenvector number
2×10^{-2}	2	D^2	16
2×10^{-3}	4	D^4	12
2×10^{-6}	8	D^8	11
2×10^{-8}	8	D^8	16
2×10^{-11}	16	D^{16}	11
2×10^{-13}	16	D^{16}	11

TABLE 4. ERROR PROPAGATION FOR THE CASE p = 4

Tolerance = $.2 \times 10^{-5}$

s	$(\frac{\Lambda_{s-1}}{\Lambda_s})^4$	e_{s-1}	ε_{s+1}
1	--	--	.00000021
2	7.02	.00000146	.00000029
3	6.54	.00000191	.00000100
4	2.16	.00000217	.00000065
5	2.48	.00000161	.00001051
6	1.15	.00001212	.00000074
7	2.43	.00000181	.00000073
8	1.82	.00000132	.00000400
9	1.40	.00000559	.00000436
10	1.09	.00000475	.00000068
11	2.16	.00000146	.00000119
12	1.53	.00000183	.00000383
13	1.22	.00000465	.00000060
14	1.84	.00000110	.00004045
15	1.02	.00004131	.00000020
16	2.13	.00000042	.00000542
17	1.08	.00000587	.00000079
18	1.67	.00000132	.00000044
19	1.61	.00000071	.00001817
20	1.01	.000001841	.00000025
21	1.85	.00000046	.00000451
22	1.10	.00000493	.00000021
23	1.80	.00000038	.00000035
24	1.60	.00000056	.00000036
25	1.63	.00000059	--

TABLE 5. EIGENVALUES AND FREQUENCIES FOR THE BEAM PROBLEM OF 50 D.O.F.

p = 1 Tolerance = $.2 \times 10^{-8}$

Mode number	frequency given by (3.15)	computed frequency	eigenvalue λ_i	$\frac{\lambda_i}{\lambda_{i+1}}$
1	.0312	.0312	1027.28	16.00
2	.1248	.1248	64.20	5.07
3	.2809	.2809	12.67	3.16
4	.4994	.4994	4.01	2.44
5	.7803	.7803	1.642	2.07
6	1.1236	1.1238	.792	1.85
7	1.5293	1.5299	.427	1.71
8	1.9975	1.9988	.250	1.60
9	2.5280	2.5308	.156	1.52
10	3.1210	3.1262	.102	1.47
11	3.7765	3.7855	.0692	1.42
12	4.4943	4.5094	.0492	1.38
13	5.2746	5.2988	.0356	1.35
14	6.1172	6.1546	.0264	1.32
15	7.0223	7.0781	.0200	1.30
16	7.9899	8.0709	.0153	1.28
17	9.0198	9.1346	.0120	1.26
18	10.1122	10.2713	.0095	1.25
19	11.2670	11.4830	.0076	1.24
20	12.4847	12.7718	.0061	1.23
21	13.7638	14.1392	.0050	1.21
22	15.1058	15.5846	.0041	1.20
23	16.5103	17.0998	.0034	1.19
24	17.9772	18.6330	.0029	1.35
25	19.5065	21.6506	.0022	1.07
26	21.0983	22.4210	.0019	1.17
27	22.7524	24.2697	.0017	1.11
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
43	57.7081	62.5020	.0026	1.03
44	60.4234	63.6678	.0025	1.19
45	63.2011	69.5303	.00022	--
46	66.0413	FAILURE		
47	68.9438			
48	71.9088			
49	74.9362			
50	78.0261			

TABLE 6. NUMBER OF ITERATIONS FOR DIFFERENT VALUES OF p

Tolerance = $.2 \times 10^{-8}$

eigenvector number	D	D^2	D^3	D^4	D^8	D^{16}
1	6	4	3	3	2	2
2	36	18	12	9	19	FAILURE
3	11	6	5	4	FAILURE	
4	59	28	18	14		
5	16	9	7	FAILURE		
6	82	36	21			
7	21	11	14			
8	103	45	114			
9	26	14	FAILURE			
10	123	54				
11	31	16				
12	140	58				
13	35	19				
14	159	62				
15	40	21				
16	176	67				
17	44	23				
18	194	71				
19	48	25				
20	202	73				
21	51	28				
22	219	78				
23	27	38				
24	76	97				
25	220	FAILURE				
26	257					
27	64					
.	.					
.	.					
.	.					
43	90					
44	389					
45	304					
46	FAILURE					
47						
48						
48						
50						

4. SUMMARY

In the first part of this paper a modification of matrix iteration using the power method is proposed. By counting operations it is shown that the modification is more efficient when the problem parameter R, Eq. (1.24), is large and the optimum value of the exponent p, Eq. (1.12), is related to the same parameter R.

A first-order error analysis demonstrates that the error is related to the ratio of adjacent eigenvalues raised to the power p. Thus, the method will behave well only for relatively small values of p.

Two numerical examples, one involving a membrane and the other involving a beam, are used to test the method under different circumstances. In the membrane problem, where the ratio of adjacent frequencies is small, we can save approximately 75% of the computational work using p=4. In the beam problem, where the frequencies are more widely spaced, we cannot use a value of p larger than 2 if we are to obtain as many as 24 modes. For p=2, we can economize approximately 50% of the computational effort.

For practical implementation of the method, one should start conservatively with p=2. If, after computing the first three or four frequencies it is found that the frequencies are not very closely spaced, then the process could be carried out using p=4. Moreover, if we are interested in finding a relatively small number of modes, the value used for p could be higher.

REFERENCES

1. L. Meirovitch, Computational Methods in Structural Dynamics, Sijthoff-Noordhoff, The Netherlands, 1980.

2. J. N. Franklin, Matrix Theory, Prentice-Hall, Englewood Cliffs, New Jersey, 1968.

3. K. Bathe, Numerical Methods in Finite Element Analysis, Prentice Hall, Englewood Cliffs, New Jersey, 1976.

4. M. Paz, Structural Dynamics - Theory and Computation, Van Nostrand Reinhold, 1980.

The vita has been removed from the scanned document