

## AN ACCELERATED SUBSPACE ITERATION METHOD

Klaus-Jürgen BATHE and Seshadri RAMASWAMY  
*Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.*

Received 17 July 1979

The subspace iteration method for solving symmetric eigenproblems in computational mechanics is considered. Effective procedures for accelerating the convergence of the basic subspace iteration method are presented. The accelerated subspace iteration method has been implemented and the results of some demonstrative sample solutions are presented and discussed.

### 1. Introduction

The analysis of a number of physical phenomena requires the solution of an eigenproblem. It is therefore natural that with the increased use of computational methods operating on discrete representations of physical problems the development of efficient algorithms for the calculation of eigenvalues and eigenvectors has attracted much attention [1]–[8]. In particular, the use of finite element and finite difference techniques on the digital computer can lead to large systems of equations, and the efficiency of an overall response analysis can depend to a significant degree on the effectiveness of the solution of the required eigenvalues and vectors.

In this paper we consider the solution of the smallest eigenvalues and corresponding eigenvectors of the generalized eigenproblem arising in dynamic analysis:

$$\mathbf{K}\boldsymbol{\phi} = \lambda\mathbf{M}\boldsymbol{\phi}, \quad (1)$$

where  $\mathbf{K}$  and  $\mathbf{M}$  are the stiffness and mass matrices of the discrete degree of freedom system, and  $(\lambda_i, \boldsymbol{\phi}_i)$  is the  $i$ th eigenpair. If the order of  $\mathbf{K}$  and  $\mathbf{M}$  is  $n$ , we have  $n$  eigenpairs which we order as follows:

$$0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_n, \\ \boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \boldsymbol{\phi}_3, \dots, \boldsymbol{\phi}_n. \quad (2)$$

The solution for the lowest  $p$  eigenvalues and corresponding eigenvectors can be written as

$$\mathbf{K}\boldsymbol{\Phi} = \mathbf{M}\boldsymbol{\Phi}\boldsymbol{\Lambda}, \quad (3)$$

where the columns of  $\boldsymbol{\Phi}$  contain the required eigenvectors, and  $\boldsymbol{\Lambda}$  is a diagonal matrix with

the eigenvalues on its diagonal:

$$\Phi = [\phi_1, \dots, \phi_p], \quad A = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_p \end{bmatrix}. \quad (4)$$

It should be noted that the eigenproblem given in eq. (1) also arises in heat transfer analysis, the analysis of associated field problems and buckling analysis.

Among the techniques for calculating the lowest  $p$  eigenvalues and corresponding eigenvectors of eq. (1) the subspace iteration method has proven to be efficient. This solution method—referred to in this paper as the *basic* subspace iteration method—consists of the following three steps [3], [7], [10]:

- Step (1).* Establish  $q$  starting iteration vectors,  $q > p$ , which span the starting subspace  $E_1$ .
- Step (2).* Perform subspace iterations, in which simultaneous inverse iteration is used on the  $q$  vectors, and Ritz analysis is employed to extract optimum eigenvalue and eigenvector approximations at the end of each inverse iteration.
- Step (3).* After iteration convergence use the Sturm sequence check to verify that the required eigenvalues and corresponding eigenvectors have been obtained.

Considering step (1), the variable  $q$  is input by the user or  $q = \min\{2p, p + 8\}$ , and the starting iteration vectors are established as discussed in [7] or by using the Lanczos algorithm. Both procedures are briefly summarized in appendix A.

Consider next step (2) and store the starting iteration vectors in  $X_1$ . The subspace iterations are performed as follows:

For  $k = 1, 2, \dots$  iterate from subspace  $E_k$  to subspace  $E_{k+1}$ :

$$K\bar{X}_{k+1} = MX_k. \quad (5)$$

Calculate the projections of the matrices  $K$  and  $M$  onto  $E_{k+1}$ :

$$K_{k+1} = \bar{X}_{k+1}^t K \bar{X}_{k+1}, \quad (6)$$

$$M_{k+1} = \bar{X}_{k+1}^t M \bar{X}_{k+1}. \quad (7)$$

Solve for the eigensystem of the projected matrices:

$$K_{k+1}Q_{k+1} = M_{k+1}Q_{k+1}A_{k+1}. \quad (8)$$

Calculate an improved approximation to the eigenvectors:

$$X_{k+1} = \bar{X}_{k+1}Q_{k+1}. \quad (9)$$

Then, provided that the iteration vectors in  $X_1$  are not orthogonal to one of the required eigenvectors (and assuming an appropriate ordering of the vectors), we have that the  $i$ th diagonal entry in  $A_{k+1}$  converges to  $\lambda_i$  and the  $i$ th vector in  $X_{k+1}$  converges to  $\phi_i$ . In this iteration the ultimate rate of convergence of the  $i$ th iteration vector to  $\phi_i$  is  $\lambda_i/\lambda_{q+1}$ , and the ultimate rate of convergence to the  $i$ th eigenvalue is  $(\lambda_i/\lambda_{q+1})^2$ . In the iteration, convergence is

measured on the eigenvalue approximations [7, p. 504],

$$tolc = \frac{|\lambda_i^{(k+1)} - \lambda_i^{(k)}|}{\lambda_i^{(k+1)}}, \quad (10)$$

where for convergence  $tolc$  must be smaller than  $tol$ . This final convergence tolerance  $tol$  is typically equal to  $10^{-6}$ , which yields a stable eigensolution and sufficient accuracy in the calculated eigenvalues and eigenvectors for practical analysis [7].

In this basic subspace iteration method, convergence has been achieved if  $tolc \leq tol$  for  $i = 1, \dots, p$  and the Sturm sequence check is passed. Considering the Sturm sequence check in step (3) above, the procedure to apply this check has been described in detail in [7]. The Sturm sequence check is very important in that it is the only means to make sure that indeed the required number of eigenpairs has been evaluated.

Considering the solution of problems for a relatively large number of eigenpairs, say  $p > 50$ , experience shows that the cost of solution using the above basic subspace iteration method rises rapidly as the number of eigenpairs considered is increased. This rapid increase in cost is due to a number of factors that can be neglected when the solution of only a few eigenpairs is required. An important point is that a relatively large number of subspace iterations may be required if the default value for  $q$  given above is employed. Namely, in this case, when  $p$  is large, the convergence rate to  $\phi_p$ , equal to  $\lambda_p/\lambda_{q+1}$ , can be close to one. On the other hand, if  $q$  is increased, the numerical operations per subspace iteration are increased significantly. Another shortcoming of the basic subspace iterations with  $q$  large is that a relatively large number of iteration vectors is used throughout *all* subspace iterations in eqs. (5) to (9). Namely, convergence to the smallest eigenvalues is generally achieved in only a very few iterations, and the converged vectors plus the  $(p+1)$ st to  $q$ th iteration vectors are only included in the additional iterations to provide solution stability and to accelerate the convergence to the larger required eigenvalues. A further important consideration pertains to the high-speed core and low-speed back-up storage requirements. As the number of iteration vectors  $q$  increases, the number of matrix blocks that need be used in an out-of-core solution can also increase significantly and the peripheral processing expenditures can be large. Finally, it is noted that the number of numerical operations required in the solution of the reduced eigenproblem in eq. (8) becomes significant when  $q$  is large and cannot be neglected in the operation count of the subspace iteration method. For these reasons the operation count given in [7, p. 507] is not applicable when  $q$  is large.

The above brief discussion shows that modifications to increase the effectiveness of the basic subspace iteration procedure are very desirable, in particular when the solution of a large number of eigenpairs is considered. The development of acceleration procedures to the subspace iteration method has been the subject of some earlier research [9]–[11]. In principle, a number of techniques can be employed, such as Aitken acceleration, overrelaxation, the use of Chebyshev polynomials and shifting; however, the difficulty is to provide a reliable and significantly more effective solution method, and such technique has not as yet been presented.

The objective in this paper is to describe an accelerated subspace iteration method that is reliable and significantly more effective than the basic scheme. We first discuss the theory and implementation of the acceleration procedures employed. These acceleration schemes have

been implemented and used in the solution of a large number of problems. To demonstrate the basic features of the solution method, we present some solution results in the paper and compare these with the results obtained using the determinant search method and the basic subspace iteration method [7]. We conclude that the new accelerated subspace iteration solution technique represents a very significant extension of the basic subspace iteration method.

## 2. Overrelaxation of iteration vectors

Overrelaxation techniques are commonly employed in iterative solution methods, and it can be expected that overrelaxation is also useful in the subspace iteration solution of eigenproblems. To incorporate overrelaxation into the subspace iterations, eqs. (5) to (8) remain unaltered, but the new iteration vectors  $\mathbf{X}_{k+1}$  are obtained using instead of (9) the relation

$$\mathbf{X}_{k+1} = \mathbf{X}_k + (\bar{\mathbf{X}}_{k+1} \mathbf{Q}_{k+1} - \mathbf{X}_k) \boldsymbol{\alpha}, \quad (11)$$

where  $\boldsymbol{\alpha}$  is a diagonal matrix with its diagonal elements equal to individual vector overrelaxation factors  $\alpha_i$ ,  $i = 1, \dots, q$ , which are calculated as discussed below.

### 2.1. Preliminary considerations on vector overrelaxation

The use of overrelaxation of an iteration vector assumes that the vector has settled down and reached its asymptotic convergence rate. The overrelaxation factor is a function of this rate of convergence, and if the overrelaxation factor is chosen based on  $\lambda_{q+1}$ , the analysis in [10] gives

$$\alpha_i = \frac{1}{1 - \lambda_i / \lambda_{q+1}}. \quad (12)$$

It is therefore necessary to have a reliable scheme for the calculation of the vector convergence rate  $\lambda_i / \lambda_{q+1}$ . Such a scheme is the essence of our method of overrelaxation.

### 2.2. The overrelaxation method used

Assuming that some of the iteration vectors have reached their asymptotic rate of convergence and we have a reasonable approximation to the corresponding eigenvalues, our objective is to calculate an approximation to  $\lambda_{q+1}$ , so that eq. (12) can be employed to evaluate the overrelaxation factors. The approximation to  $\lambda_{q+1}$  is calculated effectively using the successive eigenvalue predictions obtained during the subspace iterations.

Considering the convergence to  $\lambda_i$ , let

$$r_i^{(k+1)} = \frac{\lambda_i^{(k+1)} - \lambda_i}{\lambda_i^{(k)} - \lambda_i}, \quad (13)$$

where thus

$$\lim_{k \rightarrow \infty} r_i^{(k+1)} = \left( \frac{\lambda_i}{\lambda_{q+1}} \right)^2. \quad (14)$$

Then we can say that, approximately,

$$r_i^{(k+1)} = \frac{|\lambda_i^{(k+1)} - \lambda_i^{(k)}|}{|\lambda_i^{(k)} - \lambda_i^{(k-1)}|}, \quad i = 1, \dots, p. \quad (15)$$

Depending on the iteration number, the convergence rate estimates in eq. (15) can be grossly in error and, due to finite precision arithmetic, will certainly be meaningless at or near convergence. However, the estimates are fairly reliable if the following two conditions are satisfied:

$$\left| \frac{r_i^{(k+1)} - r_i^{(k)}}{r_i^{(k+1)}} \right| \leq \text{tolr} \quad \text{and} \quad 10^{-3} \leq \text{tolc} \leq 10^{-10}, \quad (16)$$

where *tolr* is typically 0.2 to 0.35, and *tolc* is defined in eq. (10). In using the above tolerances and all tolerances specified in the discussion to follow, we assume that a computer with 14 or more digit arithmetic is employed (e.g. CDC machines in single precision arithmetic, IBM and UNIVAC machines in double precision arithmetic).

Assume that in iteration *l* there are some eigenvalue estimates (of the *p* eigenvalues to be calculated) for which the tests in eq. (16) are passed; then, using each of these eigenvalue estimates, we can calculate an approximation to  $\lambda_{q+1}$ :

$$\lambda_{q+1} \doteq \frac{\lambda_i^{(l+1)}}{\sqrt{r_i^{(l+1)}}} \quad (17)$$

and use as the best estimate for  $\lambda_{q+1}$  the average  $\tilde{\lambda}_{q+1}$  of all estimates ever calculated in the iterations.

The value  $\tilde{\lambda}_{q+1}$  is employed instead of  $\lambda_{q+1}$  in eq. (12) to evaluate the overrelaxation factor  $\alpha_i$ .

### 3. Acceleration through shifting

The basic premise of using shifting procedures in the subspace iterations is that the rates of convergence to the required eigenpairs can be increased significantly and to such an extent as to outpay the added computational expense that is due to the additional triangular factorizations. It was shown earlier, when considering the solution of only a few eigenpairs, that for small-banded systems the determinant search algorithm is more effective than the basic subspace iteration method [7], and that for large-banded systems the subspace iteration method is more efficient. It can therefore be conjectured that, depending on the bandwidth of the system matrices and the number of eigenpairs to be calculated, some shifting procedure

that is similar to the one employed in the determinant search method should be effective in the subspace iteration method.

### 3.1. Preliminary considerations on matrix shifting

Considering shifting in the subspace iterations, it is most important to develop a stable and reliable solution scheme. A major difficulty in the use of shifting is that if a shift is on or very close to an eigenvalue, all iteration vectors immediately converge to the eigenvector corresponding to that eigenvalue. The vectors can then not be orthogonalized any more and the iteration is unstable. If the shift is very close to an eigenvalue, the last pivot element in the **LDL'** factorization of the coefficient matrix is small (compared to its original value) and the shift must be changed, but two serious situations can arise that are described qualitatively as follows:

- (1) If a shift is close but not very close to an eigenvalue (which is a situation in between the case of a shift "exactly" on an eigenvalue and the case of a shift far away from an eigenvalue), the attraction of an iteration vector to the shift may just be counter-balanced by the vector orthogonalization process. In such case, if the convergence tolerance employed is not high enough, an iteration vector is erroneously considered to have converged to an eigenvector.
- (2) Although an iteration vector may have converged already to a required eigenvector, if a shift is imposed, and this iteration vector is still included in the subspace iterations, it is possible that this iteration vector may deteriorate and suddenly converge to another eigenvector.

With due regard to these difficulties the shifting procedure presented below is a simple and stable algorithm to accelerate the convergence of the subspace iterations.

### 3.2. The shifting procedure used

Assume that the smallest  $r$  eigenvalues have already converged, i.e. we have  $tolc \leq tol$  (using eq. (10)) for the approximations to the  $r$  smallest consecutive eigenvalues. The calculated  $r$  eigenvalue approximations the estimate for  $\lambda_{q+1}$  defined in section 2.2 as  $\tilde{\lambda}_{q+1}$  and the eigenvalue iterates that are converging to the higher eigenvalues ( $i > r$ ) and satisfy eq. (10) with  $tolc \leq 10^{-2}$  are employed to establish an appropriate algorithm for shifting in the subspace iterations.

In order that the iteration vectors continue to converge monotonically to the required  $p$  eigenvectors, the shift  $\mu_s$  must satisfy the following condition:

$$\mu_s - \lambda_1 < \lambda_{q+1} - \mu_s, \quad (18)$$

which means that  $\mu_s$  is in the left half of the eigenvalue spectrum  $\lambda_1$  to  $\lambda_{q+1}$ . After shifting to  $\mu_s$  the new convergence rates to the eigenvectors are  $|\lambda_i - \mu_s|/|\lambda_{q+1} - \mu_s|$ . To satisfy eq. (18) in a conservative manner, we use

$$\mu_s \leq \bar{\lambda}_1 + \frac{1}{3}(\tilde{\lambda}_{q+1} - \bar{\lambda}_1), \quad (19)$$

where  $\bar{\lambda}_1$  is the calculated approximation to  $\lambda_1$ .

A second requirement for continued stability of the subspace iterations is that  $\mu_s$  must be relatively far away from an eigenvalue. The shift  $\mu_s$  is therefore chosen to lie midway between two well-spaced eigenvalue approximations, and these iterates must have converged to a tolerance (*tolc*) equal to  $10^{-10}$  in eq. (10).

The criterion for shifting in the iterations is that the improvement in the convergence to the higher eigenvalues must outweigh the computational expense of performing a shift. Therefore, a shift is performed if the following condition is met:

$$\left[ \begin{array}{l} \text{number of operations to} \\ \text{perform the shift} \end{array} \right] + \left[ \begin{array}{l} \text{number of operations to} \\ \text{obtain convergence} \\ \text{after shifting} \end{array} \right] < \left[ \begin{array}{l} \text{number of operations to} \\ \text{obtain convergence} \\ \text{without shifting} \end{array} \right]. \quad (20)$$

Let  $\lambda_i^{(k+1)}$  be the latest estimate of  $\lambda_i$  with *tolc*  $\leq 10^{-2}$  in eq. (10), where  $r < i \leq p$ , and let  $\tilde{\lambda}_{q+1}$  be the latest estimate for  $\lambda_{q+1}$ ; then we can estimate the convergence rate to  $\lambda_i$  as  $d = (\lambda_i^{(k+1)}/\tilde{\lambda}_{q+1})^2$ . Also, let  $t$  be the number of additional iterations to reach convergence without shifting. We can estimate  $t$  using

$$d' = \text{tol}i, \quad (21)$$

where *tol**i* is the increase in accuracy still to be gained in the subspace iterations; here we have *tol**i* = *tol*/*tolc*. Hence

$$t = \frac{\log(\text{tol}i)}{\log(d)}. \quad (22)$$

Using the information in eqs. (20)–(22), we can evaluate whether it is efficient to shift. Assume that a stable shift is given by  $\mu_s$  (i.e.  $\mu_s$  is chosen using eq. (19)); then the convergence rate  $\bar{d}$  to  $\lambda_i$  after shifting would be approximately

$$\bar{d} = \frac{(\lambda_i^{(k+1)} - \mu_s)^2}{(\lambda_{q+1} - \mu_s)^2}. \quad (23)$$

Hence the number of subspace iterations required for convergence after shifting are approximately

$$\bar{t} = \frac{\log(\text{tol}i)}{\log(\bar{d})}. \quad (24)$$

The above values for  $t$  and  $\bar{t}$  are calculated for all eigenvalue iterates  $\lambda_i^{(k+1)}$ , where  $r < i \leq p$ . The maximum difference between  $t$  and  $\bar{t}$ , given by  $(t - \bar{t})_{max}$ , is then used in eq. (20). Thus a

shift is efficient if (see [7, table 12.3, p. 507])

$$\begin{aligned} \frac{1}{2} nm^2 &< \{n(2qm + 2q^2) + 18q^3\}(t - \bar{t})_{max} && \text{(lumped mass matrix),} \\ \frac{1}{2} nm^2 &< \{n(4qm + 2q^2) + 18q^3\}(t - \bar{t})_{max} && \text{(banded mass matrix),} \end{aligned} \quad (25)$$

where  $m$  is the average bandwidth of the stiffness matrix in eq. (1),  $m_M = 0$  for a lumped mass idealization, and  $m_M = m$  for a consistent mass idealization.

In the operation count shown in eq. (25) it is assumed that all  $q$  iteration vectors are included in the complete subspace iterations. Eq. (25) is modified if some of the iteration vectors are accurately approximating respective eigenvectors, i.e. in eq. (10)  $tolc \leq 10^{-10}$ . In such case we accept the iteration values as fully converged and, corresponding to the converged vectors, we dispense with additional inverse iterations in eq. (5), the projection calculations in eqs. (6) and (7) and the calculation of new iteration vectors in eq. (9). Apart from saving some numerical operations, by not performing these calculations we also safeguard against a possible loss of accuracy in the already converged vectors during further subspace iterations at new shifts.

### 3.3. Iteration procedure for the case $q < p$

In the previous sections we discussed the acceleration of the basic subspace iteration scheme by shifting and overrelaxation when throughout the solution the number of iteration vectors is significantly larger than  $p$ . However, in some cases, when  $p$  is large, the effectiveness of the solution is increased if the required  $p$  eigenpairs are calculated with a number of iteration vectors  $q$  smaller than  $p$ . Basically, two advantages can arise in such solution. Firstly, the required high speed core storage decreases, and, secondly, unnecessary orthogonalizations of iteration vectors to already calculated eigenvectors are automatically avoided.

In case  $q < p$  the solution algorithm proceeds as described in section 3.2 with the following modifications (see fig. 1). Since the required eigenpairs cannot be calculated effectively without shifting, and a shift should lie in-between eigenvalue iterates that have converged to a tolerance of  $10^{-10}$  (using eq. (10)), the decision on whether to shift is based on the eigenvalue iterates that have not yet converged to the  $10^{-10}$  tolerance. Hence, in the analysis of section 3.2 the variable  $r$  is equal to the number of consecutive smallest eigenvalue iterates that have all converged to  $tolc \leq 10^{-10}$ .

Assume next that eq. (25) shows that  $\mu_s$  should be increased beyond the limit given by eq. (19). In this case a shift is not performed, but the iteration vectors that correspond to the smallest eigenvalue iterates and that consecutively all satisfy eq. (10) to a tolerance of  $10^{-10}$  are transferred to back-up storage and replaced with new starting iteration vectors. The effect of this procedure is to increase  $\bar{\lambda}_{q+1}$ , by which the rate of convergence of the iteration vectors is improved, and also allow further shifting. Considering the next subspace iterations, it is important to assure that the iteration vectors do not converge to already calculated eigenvectors, and it is effective to employ Gram-Schmidt orthogonalization. All  $q$  iteration vectors are orthogonalized prior to a subspace iteration to the eigenvectors  $\phi_i$  that have already been stored on a back-up storage and correspond to eigenvalues  $\lambda_i$  with

$$|\bar{\lambda}_i - \mu_s| \leq |\lambda^* - \mu_s|, \quad (26)$$



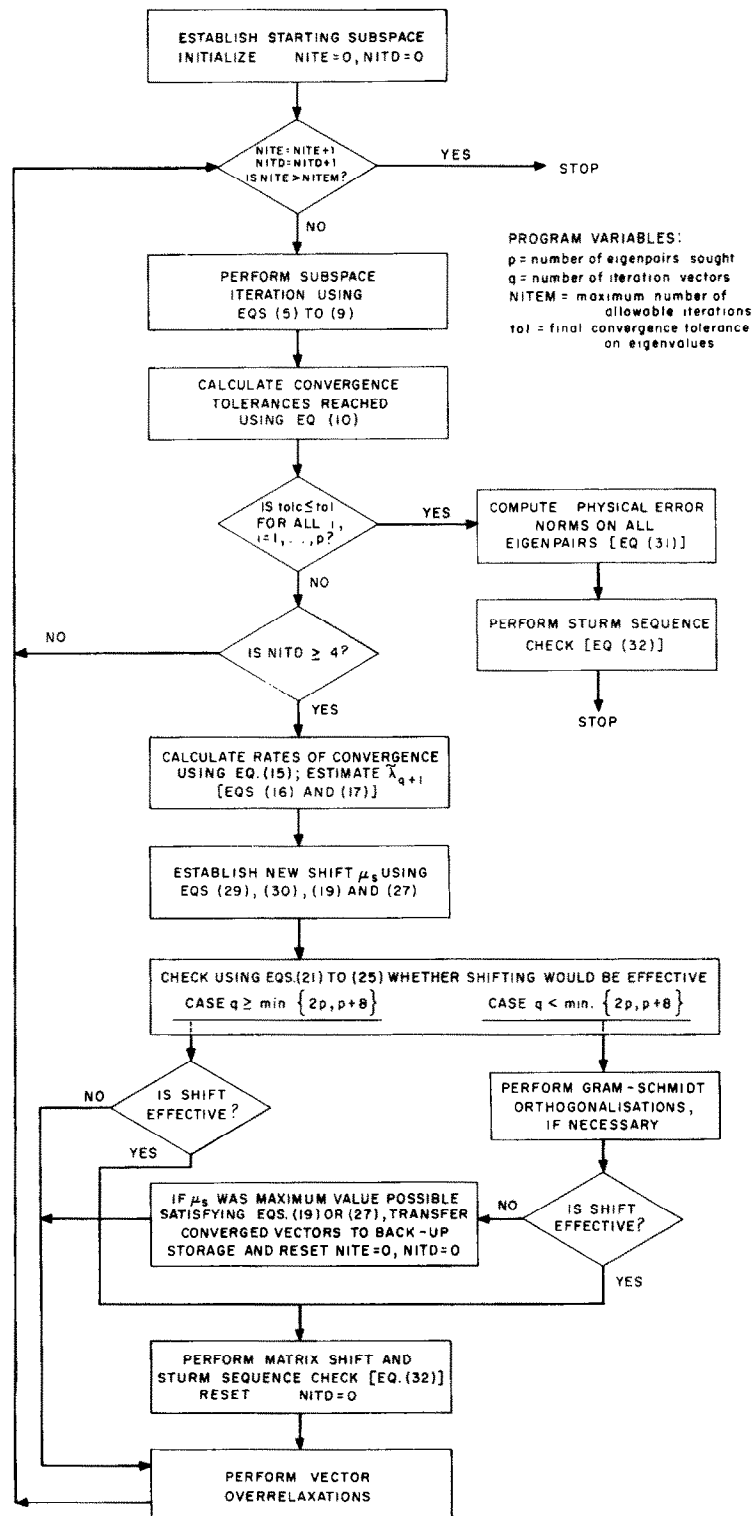


Fig. 1. Calculations in accelerated subspace iteration method.

where  $\lambda^*$  is the largest eigenvalue estimate calculated in the current subspace iteration, and  $\bar{\lambda}_i$  represents the calculated approximation to  $\lambda_i$ . The number of Gram-Schmidt orthogonalizations thus performed is conservative.

Considering the selection of additional shifting, we note that once accurate approximations to the eigenvectors  $\phi_1, \dots, \phi_j$  have been calculated and stored on back-up storage, the smallest eigenvalue to be calculated next is  $\lambda_{j+1}$ , and hence eq. (19) is modified to

$$\mu_s \leq \bar{\lambda}_{j+1} + \frac{1}{3}(\bar{\lambda}_{q+j+1} - \bar{\lambda}_{j+1}), \quad (27)$$

where  $\bar{\lambda}_{j+1}$  is the calculated approximation to  $\lambda_{j+1}$ .

This shifting strategy is used effectively if  $q < p$ , but to avoid convergence difficulties, the strategy is also best employed if  $q$  is only slightly larger than  $p$ . We thus recommend using the iteration vector replacement strategy when  $q$  is smaller than the default value quoted in section 1, i.e. when  $q < \min\{2p, p + 8\}$ .

#### 4. Computer implementation

The solution scheme presented in the previous section has been implemented in the computer program SSPACE [7, p. 509]. The purpose of this section is to summarize how the solution procedures have been implemented in the program (see fig. 1).

##### 4.1. Overrelaxation

Having calculated  $X_{k+1}$  in eq. (9), an overrelaxation is performed for an iteration vector if eq. (16) is satisfied. In the overrelaxation the iteration vectors from the previous iteration are read from back-up storage, and we calculate

$$X_{k+1} \leftarrow X_k + (X_{k+1} - X_k)\alpha. \quad (28)$$

##### 4.2. Shifting

Considering the matrix shifting strategy, the initial shift is zero and then  $\mu_s$  is increased with each shift. Based on the considerations in section 3.2, we assess after 4 subspace iterations at a shift whether it is effective to shift to a new position. Namely, about 4 subspace iterations are required for the iteration vectors to settle down after a shift, so that the analysis presented above is approximately applicable. Also, a shift is performed only if there is a saving of at least 3 subspace iterations, i.e. in addition to satisfying eq. (25), we must also have  $(t - \bar{t})_{max} \geq 3$ .

After 4 subspace iterations have been performed at the current shift the following procedure is employed to establish a *new* shift  $\mu_s$ .

- (a) Calculate the convergence rates  $r_i$  of the eigenvalue iterates using eq. (15) at the end of each iteration, and calculate  $\bar{\lambda}_{q+1}$  using eqs. (15) and (17).

(b) Establish the largest allowable shift  $\mu_s$ . This shift is calculated as

$$\mu_s = \frac{\bar{\lambda}_s + \bar{\lambda}_{s-1}}{2}, \quad (29)$$

where  $\bar{\lambda}_s$  is the calculated approximation to  $\lambda_s$  and  $\lambda_s$  is the largest eigenvalue for which all eigenvalue iterates, below and including  $\lambda_s$ , have converged to a tolerance of  $10^{-10}$  using eq. (10). Check whether this shift satisfies eq. (19) (or eq. (27)) and also the condition

$$1.01\bar{\lambda}_{s-1} \leq \mu_s \leq 0.99\bar{\lambda}_s. \quad (30)$$

If either eq. (19) (or eq. (27)) or eq. (30) is not satisfied, decrease  $s$  (using  $s \leftarrow s - 1$ ) until both conditions are met. It is next assessed whether shifting to  $\mu_s$  is effective if the value of  $\mu_s$  thus obtained is still larger than the current shift.

- (c) If only a few subspace iterations have been performed, reasonably accurate estimates for all  $\lambda_m$ ,  $s < m \leq p$ , may not yet be attainable. Hence, to evaluate eqs. (21)–(25), we use only the eigenvalue iterates  $\bar{\lambda}_m$  for which  $tolc \leq 10^{-2}$ . In order that shifting to  $\mu_s$  be performed, eq. (25) must be satisfied.
- (d) If a shift is performed, use the Sturm sequence information and error estimates on the calculated eigenpair approximations to establish whether all eigenvalues between the previous shift and the new shift have been obtained [7, p. 505]. Assume that  $j$  eigenvalues have been calculated between the previous and the current shift; then the following physical error norms [7, p. 413] should be small for all eigenpairs calculated:

$$\epsilon_i = \frac{\|\mathbf{K}\phi_i^{(i+1)} - \lambda_i^{(i+1)}\mathbf{M}\phi_i^{(i+1)}\|_2}{\|\mathbf{K}\phi_i^{(i+1)}\|_2}, \quad (31)$$

and  $j$  additional negative elements must be measured in  $\mathbf{D}$ , where

$$\mathbf{K} - \mu_s\mathbf{M} = \mathbf{LDL}^t. \quad (32)$$

In theory, it could happen that an eigenpair has been missed [7, p. 505]. However, in practice, such a situation is extremely rare and would always be detected; therefore, the solution procedure is a reliable analysis tool. Also, because the missing of an eigenpair is so rare, the recommended remedy is somewhat crude; namely, stop the solution and repeat with a larger number  $q$  of iteration vectors and possibly a tighter convergence tolerance  $tol$  [7].

Considering the case  $q < \min\{2p, p + 8\}$ , the matrix shifting strategy is as described above with one additional calculation procedure. Assume that the candidate shift is discarded based on eq. (25) and is the maximum value possible satisfying eq. (19) (or eq. (27)). In this case, all iteration vectors that correspond to the smallest eigenvalue iterates and that consecutively all satisfy eq. (10) to a tolerance of  $10^{-10}$  are written on back-up storage and replaced by new starting iteration vectors. Further checking on whether additional matrix shifting is effective is then performed after four more subspace iterations.



Table 1. Analysis of the tridiagonal system ( $p = 4, q = 8$ )

	Accelerated scheme				
	Basic scheme	Standard starting subspace		Lanczos starting subspace	
		Overrelaxation only	Shifting only	Overrelaxation and shifting	Overrelaxation and shifting
Number of iterations	49	35	31	23	20

### 5.2. Analysis of a piping system

Fig. 3 shows a computer plot of the piping system considered in this study. For this system the order  $n$  of the stiffness and mass matrices is 330, the mean half-bandwidth  $m_K$  of the stiffness matrix is 26, and a diagonal mass matrix was employed. The sixty smallest eigenvalues and corresponding eigenvectors were required.

Table 2 summarizes some relevant solution data corresponding to the various analyses performed.

Considering the solution with 68 iteration vectors ( $q = 68$ ), the ratio  $(\lambda_{60}/\lambda_{69})^2$  is equal to 0.3, resulting in rapid convergence using the basic scheme. In this case there is no reduction in the number of iterations and the required high speed storage using the accelerated method. However, considering the solution with  $q = 20$  and  $q = 4$ , significantly less high speed storage is needed at no increase in central processor time. Since the average bandwidth of the stiffness matrix is small, the determinant search method is equally effective for this problem [7].

It is interesting to note that in the solution using the Lanczos starting subspace, with  $q = 68$ , after two iterations the first 34 eigenvalue iterates and after a total of only five iterations the smallest 45 eigenvalue iterates had converged to  $tolc = 10^{-10}$ . Since the converged iteration vectors are no longer included in the iterations (see section 3.2), about the same total solution

Table 2. Comparison of different solution strategies in the analysis of the piping system ( $n = 330, m_K = 26$ ). Diagonal mass matrix was used (Computer used was CDC Cyber 175,  $tol = 10^{-6}$ )

$p/q$	Accelerated scheme						
	Basic scheme	Standard starting subspace			Lanczos starting subspace		Determinant search
		60/68	60/68	60/20	60/4	60/68	
Total number of subspace iterations	13	13	74	408	23	2	(5 + 6)*
Total high speed core storage used	42,494	42,494	17,870†	11,700†	42,494	64,094	12,490†
Solution time (CPU sec)	90	74	65	63	77	71	63

\*Average number of factorizations and inverse iterations per eigenpair

†Additional secondary storage was required for storing converged eigenvectors

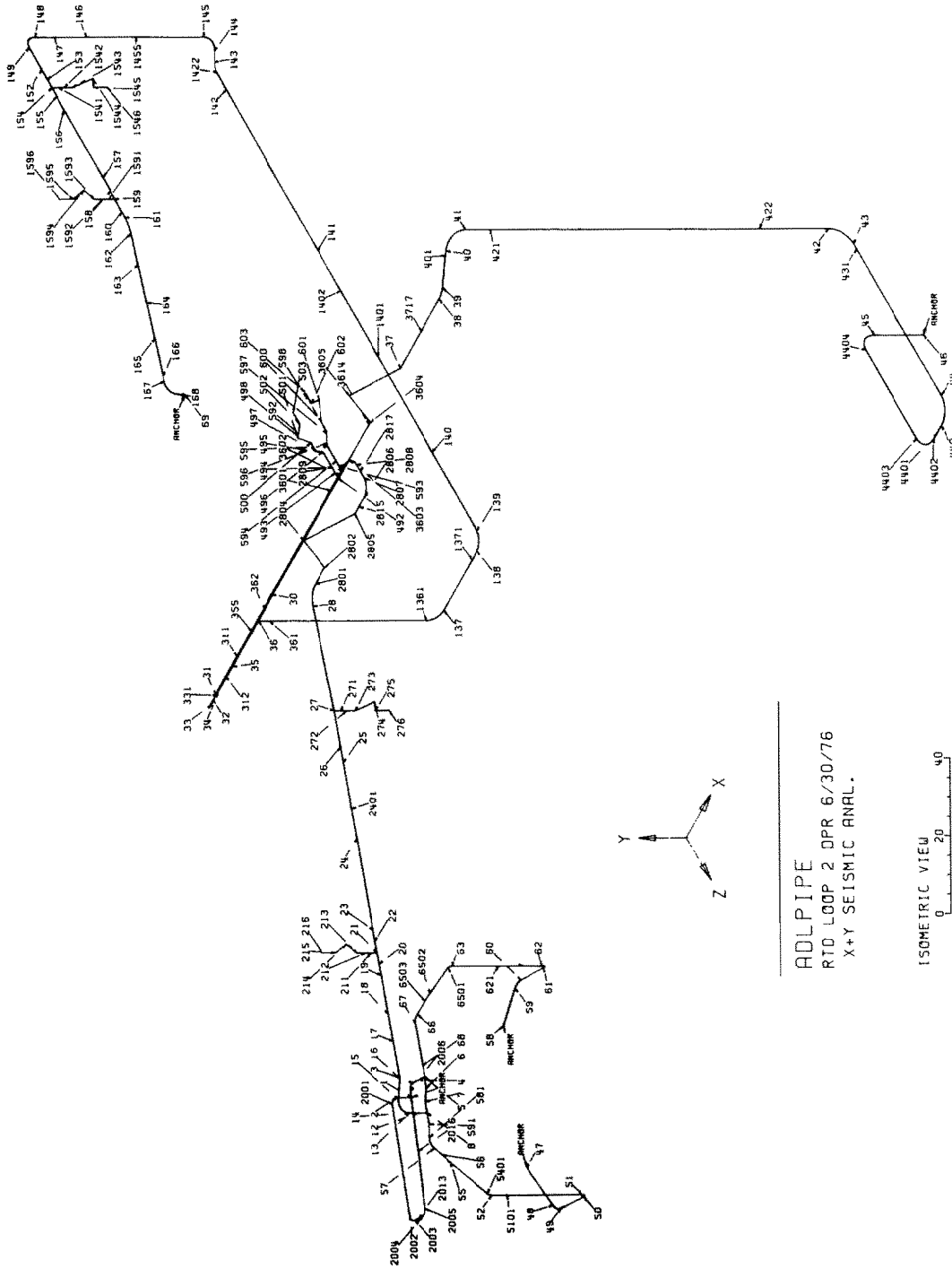


Fig. 3. Geometric view of piping system (order of matrices  $n = 330$ , mean half-bandwidth  $m_k = 26$ ).

times are required using the "standard" and the Lanczos starting subspaces although more iterations are required using the Lanczos starting vectors.

Finally, table 3 summarizes the complete solution steps for the case  $q = 20$ . As seen from this table, a total of ten matrix shifts were performed in the solution, and the required sixty eigenpairs were calculated in bundles of 15, 13, 15, 5 and 12 each from five iteration vector sets.

### 5.3. Analysis of a building frame

The building frame shown in fig. 4 was earlier analyzed in [3]. We analyzed the same structure in this study to demonstrate some important features of the accelerated subspace iteration method. For this system  $n = 468$  and  $m_K = 91$ , and a diagonal mass matrix was employed. The sixty smallest eigenvalues and corresponding eigenvectors were required.

Since the stiffness matrix has a relatively large bandwidth, a determinant search solution is not effective, and only subspace iteration solutions have been calculated.

Table 4 gives the characteristics of the solutions obtained. It is seen that the accelerated subspace iteration method yields a significantly more effective solution than the basic scheme. Table 5 summarizes the solution steps for the case  $q = 20$ .

In this analysis the Lanczos starting subspace was not employed because the stiffness matrix had to be processed in blocks due to high speed storage limitations [7]. The generation of the starting vectors using the Lanczos method in such a case requires considerable peripheral processing and is not effective when a large number of vectors need be calculated in the solution.

Table 3. Solution steps in the analysis of the piping system ( $p = 60$ ,  $q = 20$ )

Iteration vector set numbers	Eigenpairs sought	Iterations performed with these vectors	Converged trial vectors simultaneously removed to back-up storage	Eigenpair approximations carried over to next step	Matrix shifts applied at $1/2(\lambda_i + \lambda_{i-1})$ based on convergence of $\lambda_k$			Calculation of $\lambda_{q+1}$		
					Iteration no.	$i$	$k$	$q + 1$	$\tilde{\lambda}_{q+1}$ (estimated value)	$\lambda_{q+1}$
1	1-20	1-17	1-15 at iteration 17	16-20	5	1	14	21	24,501	40,373
					9	4	17		29,738	
					13	11	17		32,847	
2	16-35	18-36	16-28 at iteration 36	29-35	24	18	35	36	125,604	102,360
					28	22	35		118,106	
					32	25	35		113,883	
3	29-48	37-52	29-43 at iteration 52	44-48	40	31	44	49	182,623	254,685
					44	34	46		227,682	
					48	41	46		232,584	
4	44-63	53-66	44-48 at iteration 66	49-63	61	49	60	64	658,386	548,048
5	49-68	67-73	49-60 at iteration 74							

Table 4. Comparison of different solution strategies in the analysis of the building frame ( $n = 468$ ,  $m_k = 91$ ). Diagonal mass matrix was used (Computer used was CDC Cyber 175,  $tol = 10^{-6}$ )

$p/q$	Basic scheme		Accelerated scheme (standard starting subspace)	
Total number of subspace iterations	60/68	60/68	60/20	
Total high speed core storage used	47	36	80	
Solution time (CPU sec)	66850*	66850*	55282†	
	570	279	209	

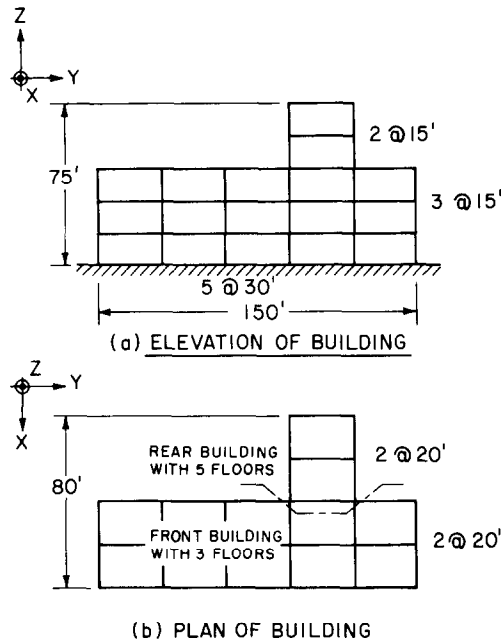
\*Stiffness matrix was stored out-of-core (4 blocks)

†Stiffness matrix was stored in-core, but additional secondary storage was required for storing converged iteration vectors

Table 5. Solution steps in the analysis of the building frame ( $p = 60$ ,  $q = 20$ )

Iteration vector set numbers	Eigenpairs sought	Iterations performed with these vectors	Converged trial vectors simultaneously removed to back-up storage	Eigenpair approximations carried over to next set	Matrix shifts applied at $1/2(\lambda_i + \lambda_{i-1})$ based on convergence of $\lambda_k$			Calculation of $\lambda_{q+1}$		
					Iteration no.	$i$	$k$	$q+1$	$\tilde{\lambda}_{q+1}$ (estimated value)	$\lambda_{q+1}$
1	1-20	1-17	1-12 at iteration 17	13-20	9	6	15	21	9.59	11.58
					13	9	16		10.01	
2	13-32	18-37	13-27 at iteration 37	28-32	21	13	24	33	17.71	29.44
					25	17	30		28.38	
					29	24	32		29.64	
					33	25	32		30.31	
3	28-47	38-54	28-42 at iteration 54	43-47	42	29	34	48	41.51	86.12
					46	34	42		55.03	
					50	41	43		65.23	
4	43-62	55-72	43-48 at iteration 72	49-62	58	43	48	63	87.64	122.63
					64	46	59		122.75	
					68	47	60		122.81	
5	49-68	73-80	49-60 at iteration 80							



**DATA**

YOUNG'S MODULUS=432000, MASS DENSITY=1.0  
 COLUMNS IN FRONT BUILDING  $A_1=3.0, I_1=I_2=I_3=1.0$   
 COLUMNS IN REAR BUILDING  $A_1=4.0, I_1=I_2=I_3=1.25$   
 ALL BEAMS INTO X-DIRECTION  $A_1=2.0, I_1=I_2=I_3=0.75$   
 ALL BEAMS INTO Y-DIRECTION  $A_1=3.0, I_1=I_2=I_3=1.0$   
 UNITS : FT, KIP

Fig. 4. A three-dimensional building frame (order of matrices  $n = 468$ , mean half-bandwidth  $m_K = 91$ ).

## 6. Conclusions

Effective strategies for accelerating the basic subspace iteration method in the calculation of the smallest eigenvalues and corresponding eigenvectors of generalized eigenproblems have been presented. The solution strategies have been implemented, and the results of some sample analyses are reported. Based on the theory used and the experience obtained with the accelerated subspace iteration method, we conclude that the technique can in some cases provide significantly more effective solutions than the basic method. The increase in solution effectiveness depends on the properties of the eigensolution sought, such as the number of eigenpairs to be calculated, the spreading of the eigenvalues and the order and bandwidths of the matrices. The accelerated solution scheme is in particular more effective than the basic subspace iteration method when the basic method converges only using relatively many iterations. Also, since the accelerated subspace iteration method can be employed with a small or large number of iteration vectors  $q$ , the method is more general than the basic method; e.g. the accelerated method can be applied effectively to the solution of eigenproblems in which the matrices have small or large bandwidths.

## 7. Acknowledgement

We would like to thank I.W. Dingwell of A.D. Little, Cambridge, Massachusetts for supplying the stiffness and mass matrices of the piping system considered in section 5.2. We are also thankful to Prof. B. Irons, University of Calgary, Canada, and Fred Peterson, Engineering/Analysis Corporation, Berkeley, California, for stimulating discussions on the subspace iteration method. Finally, we are grateful to the ADINA users group for supporting financially our research work in computational mechanics.

## Appendix. Calculation of starting iteration vectors

Two procedures have been employed to generate the starting iteration vectors.

Using the "standard" procedure, the vectors are generated as described in [7, p. 501]. Briefly, the first starting iteration vector is a full unit vector, the next  $q - 2$  vectors each are unit coordinate vectors with the unit entries corresponding to degrees of freedom with large mass and low stiffness values, and the  $q$ th starting iteration vector is a random vector. This procedure was always used in the basic subspace iteration method.

In the second procedure the Lanczos algorithm is employed to generate the starting iteration vectors [8]. This procedure is in general effective if  $q$  is considerably larger than  $p$ . Using this method, we proceed as follows:

Let

$$\bar{\mathbf{x}}^t = \{1 \quad 1 \dots 1\} \quad (\text{with all elements } 1).$$

Calculate

$$\alpha^2 = \bar{\mathbf{x}}^t \mathbf{M} \bar{\mathbf{x}},$$

and take the first starting iteration vector as

$$\mathbf{x}_1 = \bar{\mathbf{x}}/\alpha.$$

Now calculate the starting iteration vectors  $\mathbf{x}_2, \dots, \mathbf{x}_{q-1}$ , using the following equations (with  $\beta_1 = 0$ ):

$$\mathbf{K} \bar{\mathbf{x}}_{i+1} = \mathbf{M} \mathbf{x}_i, \tag{A.1}$$

$$\alpha_i = \bar{\mathbf{x}}_{i+1}^t \mathbf{M} \mathbf{x}_i, \tag{A.2}$$

$$\bar{\mathbf{x}}_{i+1} = \bar{\mathbf{x}}_{i+1} - \alpha_i \mathbf{x}_i - \beta_i \mathbf{x}_{i-1}, \tag{A.3}$$

$$\beta_{i+1}^2 = \bar{\mathbf{x}}_{i+1}^t \mathbf{M} \bar{\mathbf{x}}_{i+1}, \tag{A.4}$$

$$\mathbf{x}_{i+1} = \bar{\mathbf{x}}_{i+1} / \beta_{i+1}. \tag{A.5}$$

The  $q$ th starting iteration vector is established using a random vector and orthogonalizing this vector to all vectors  $\mathbf{x}_j$ ,  $j = 1, \dots, q - 1$ .

Theoretically, the vectors  $\mathbf{x}_i$ ,  $i = 1, \dots, q$ , that are generated by the above algorithm form an  $\mathbf{M}$ -orthonormal basis. However, in practice, the computed vectors are in general not orthogonal because of round-off errors. For this reason we orthogonalize the vectors  $\mathbf{x}_{i+1}$  obtained in eq. (A.5) to all previously computed vectors  $\mathbf{x}_j$ ,  $j = 1, \dots, i$ .

Another consideration is that the generated vector  $\mathbf{x}_{i+1}$  would theoretically be a null vector if the starting vector  $\mathbf{x}_1$  lies in an  $i$ -dimensional subspace of the operators  $\mathbf{K}$  and  $\mathbf{M}$ . Hence, we compute  $\gamma_{i+1} = (\bar{\mathbf{x}}_{i+1}^t \mathbf{M} \bar{\mathbf{x}}_{i+1})^{1/2}$ , and whenever the ratio  $\beta_{i+1}/\gamma_{i+1}$  is smaller than  $10^{-4}$ , the computed vector  $\mathbf{x}_{i+1}$  is discarded. Then we use the  $(i+1)$ st vector generated by the above "standard" procedure as the vector  $\mathbf{x}_{i+1}$ , orthogonalize it to all vectors  $\mathbf{x}_j$ ,  $j = 1, \dots, i$ , and with  $\beta_{i+1}$  equal to zero we continue the recurrence algorithm in eqs. (A.1)–(A.5).

## References

- [1] O.E. Brønlund, Eigenvalues of large matrices, Symposium on Finite Element Techniques at the Institut für Statik and Dynamik der Luft- und Raumfahrtkonstruktionen, Univ. Stuttgart, Jun. 1969.
- [2] K.K. Gupta, Solution of eigenvalue problems by the Sturm sequence method, *Int. J. Numer. Meths. Eng.* 4 (1972) 379–404.
- [3] K.J. Bathe, Solution methods for large generalized eigenvalue problems in structural engineering, Report UC SESM 71-20 (Civil Eng. Dep., Univ. California, Berkeley, CA, Nov. 1971).
- [4] P.S. Jensen, The solution of large symmetric eigenproblems by sectioning, *SIAM J. Numer. Anal.* 9 (1972) 534–545.
- [5] R.B. Corr and A. Jennings, A simultaneous iteration algorithm for symmetric eigenvalue problems, *Int. J. Numer. Meths. Eng.* 10 (1976) 647–663.
- [6] H.R. Schwarz, The eigenvalue problem  $(\mathbf{A} - \lambda \mathbf{B})\mathbf{x} = \mathbf{o}$  for symmetric matrices of high order, *Comp. Meths. Appl. Mech. Eng.* 3 (1974) 11–28.
- [7] K.J. Bathe and E.L. Wilson, *Numerical methods in finite element analysis* (Prentice-Hall, Englewood Cliffs, NJ, 1976).
- [8] C.C. Paige, Computational variants of the Lanczos method for the eigenproblem, *J. Inst. Math. Appl.* 10 (1972) 373–381.
- [9] Y. Yamamoto and H. Ohtsubo, Subspace iteration accelerated by using Chebyshev polynomials for eigenvalue problems with symmetric matrices, *Int. J. Numer. Meths. Eng.* 10 (1976) 935–944.
- [10] K.J. Bathe, Convergence of subspace iteration, in: K.J. Bathe, J.T. Oden and W. Wunderlich (eds.), *Formulations and numerical algorithms in finite element analysis* (MIT Press, Cambridge, MA, 1977).
- [11] K.J. Bathe and S. Ramaswamy, Subspace iteration with shifting for solution of large eigensystems, Report AVL 82448-7 (Dept. Mech. Eng., MIT, Cambridge, MA, Dec. 1977).
- [12] K.J. Bathe, ADINA – a finite element program for automatic dynamic incremental nonlinear analysis, Report AVL 82448-1 (MIT, Cambridge, MA, May 1975 (rev. Dec. 1978)).