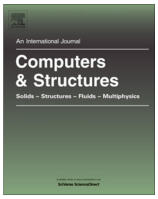
Contents lists available at SciVerse ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc



The subspace iteration method – Revisited

Klaus-Jürgen Bathe

Massachusetts Institute of Technology, Cambridge, MA, United States

ARTICLE INFO

Article history:
Received 22 May 2012
Accepted 5 June 2012
Available online 4 July 2012

Keywords:
Frequencies
Mode shapes
Subspace iteration method
Parallel processing
Shared memory
Distributed memory

ABSTRACT

The objective in this paper is to present some recent developments regarding the subspace iteration method for the solution of frequencies and mode shapes. The developments pertain to speeding up the basic subspace iteration method by choosing an effective number of iteration vectors and by the use of parallel processing. The subspace iteration method lends itself particularly well to shared and distributed memory processing. We present the algorithms used and illustrative sample solutions. The present paper may be regarded as an addendum to the publications presented in the early 1970s, see Refs. [1,2], taking into account the changes in computers that have taken place.

1. Introduction

The subspace iteration method was developed by K.J. Bathe [1] for the solution of frequencies and mode shapes of structures, and in particular for the earthquake analysis of buildings and bridges [1–3]. Originally in the 1970s, relatively few eigenpairs were sought in these solutions, like the lowest 10 to 20 frequencies and mode shapes, when the model contained a total of 1000 to 10,000 degrees of freedom. However, since its original development, the subspace iteration method has been used abundantly in research and commercial finite element programs for small and very large finite element systems, and the method has naturally attracted considerable attention for improvements, see for example Refs. [4–10].

The original development of the method given in Ref. [1] is based on vector simultaneous iterations, as proposed by Bauer [11] and Rutishauser [12], but includes the important use of the Ritz method, the selection of the iteration starting vectors, the use of an effective number of starting vectors, error measures, and the Sturm sequence check. Without the use of the Ritz step, simultaneous vector iterations are not effective. While abundantly used for frequency and linearized buckling solutions in engineering and the sciences, the method is also employed in the solution of random eigenvalue problems [13]. A convergence analysis of the subspace iteration method is given in Ref. [14].

Two attractive properties of the subspace iteration method are, firstly, its robustness and efficiency and, secondly, the fact that using a starting subspace close to the subspace of interest can lead to a very fast solution. This situation is frequently encountered in engineering and the sciences, e.g. in optimization problems and in protein dynamics. We shall focus in this paper on the selection of the number of iteration vectors and illustrate a third attractive property, namely, its use in parallelized computations.

Also, as another technique, the Lanczos method can be very effective, in particular when solving for many frequencies and mode shapes [15,16]. Initially, the Lanczos method showed instabilities due to loss of orthogonality of the iteration vectors. However, these difficulties have been largely overcome and in good implementations the method can be very efficient [17,18]. A particular asset of the method is that the computational effort may increase almost linearly with the number of eigenpairs sought. This asset can render the Lanczos method attractive compared to the original subspace iteration method if many eigenpairs need be calculated. Namely, in that case, the computational effort increases larger than linear in the *original* subspace iteration method, and this increase can be significant. The Lanczos method and Bathe's subspace iteration method (or variants of these two iterative schemes) are two techniques that, at present, are very widely used for the solution of large eigenvalue problems in finite element analysis. Any noteworthy improvements to these methods are therefore of interest.

An important step in the subspace iteration method is to establish effective starting iteration vectors, which also implies to, ideally, use the optimal number of iteration vectors.

Lately much effort has been spent on using parallel processing in finite element analysis, in shared memory and distributed memory processing modes. Whereas the Lanczos method (working on individual vectors [16]) can intrinsically only be parallelized in the

E-mail address: kjb@mit.edu

0045-7949/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compstruc.2012.06.002

factorization of the stiffness matrix and the forward reduction and back-substitution of the *individual* vectors, the subspace iteration method allows in addition the parallel solution of *multiple* iteration vectors, which can result in a large computational benefit.

In this paper we first present the subspace iteration method implying use on a single processor machine and discuss how to choose an effective number of iteration vectors in structural analyses. While the optimal number must clearly depend on the problem considered, a good choice can mean a significant reduction in computational time when many frequencies and mode shapes shall be computed. Based on the discussion, we arrive at a simple formula for the selection of a reasonable number of iteration vectors for any solution.

Thereafter, we consider the use of the subspace iteration method in parallel processing, on shared memory and distributed memory machines. In a brief discussion, we show how the method lends itself particularly well to parallel computations.

Finally, we give the results of some illustrative example solutions.

2. The basic subspace iteration method

The basic equations of Bathe's subspace iteration method have been published in Refs. [1,16], but we include them here for completeness of the presentation. Thereafter we focus on the evaluation of an effective number of iteration vectors.

2.1. The basic equations

Let $\mathbf{K}$ and $\mathbf{M}$ be the stiffness and mass matrices of a finite element system with n degrees of freedom, and consider the generalized symmetric eigenvalue problem

$$\mathbf{K}\boldsymbol{\varphi} = \lambda\mathbf{M}\boldsymbol{\varphi} \tag{1}$$

We seek the smallest p eigenvalues λ_i, $i = 1,\ldots,p$, and corresponding eigenvectors $\boldsymbol{\varphi}_i$, $i = 1,\ldots,p$, with the ordering

$$0 < \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_p \tag{2}$$

which satisfy

$$\mathbf{K}\boldsymbol{\varphi}_i = \lambda_i\mathbf{M}\boldsymbol{\varphi}_i; \quad i = 1,\ldots,p \tag{3}$$

and the Kronecker delta relationships

$$\begin{aligned} \boldsymbol{\varphi}_i^T\mathbf{M}\boldsymbol{\varphi}_j &= \delta_{ij} \\ \boldsymbol{\varphi}_i^T\mathbf{K}\boldsymbol{\varphi}_j &= \lambda_i\delta_{ij} \end{aligned} \tag{4}$$

If the smallest eigenvalue is actually equal to zero, a shift can be used to reach the situation given in Eq. (2) [16]. The basic equations used in the subspace iteration method are, for k = 1, 2, ...,

$$\mathbf{K}\overline{\mathbf{X}}_{k+1} = \mathbf{M}\mathbf{X}_k \tag{5}$$

$$\mathbf{K}_{k+1} = \overline{\mathbf{X}}_{k+1}^T\mathbf{K}\overline{\mathbf{X}}_{k+1} \tag{6}$$

$$\mathbf{M}_{k+1} = \overline{\mathbf{X}}_{k+1}^T\mathbf{M}\overline{\mathbf{X}}_{k+1} \tag{7}$$

$$\mathbf{K}_{k+1}\mathbf{Q}_{k+1} = \mathbf{M}_{k+1}\mathbf{Q}_{k+1}\boldsymbol{\Lambda}_{k+1} \tag{8}$$

$$\mathbf{X}_{k+1} = \overline{\mathbf{X}}_{k+1}\mathbf{Q}_{k+1} \tag{9}$$

In practice, it is effective to order the iteration vectors in $\mathbf{X}_k$ naturally from the first to the last columns such that these correspond to increasing eigenvalues. Then the first vector in $\mathbf{X}_k$ corresponds to the eigenvector approximation of $\boldsymbol{\varphi}_1$ and the qth vector to the eigenvector approximation of $\boldsymbol{\varphi}_q$. The calculated approximations to the eigenvalues are given in $\boldsymbol{\Lambda}_{k+1}$.

There are three distinct steps of solution.

First, the q starting iteration vectors in $\mathbf{X}_1$ are established, $q > p$, where $\mathbf{X}_1$ is a matrix of dimension $n \times q$.

Second, the iteration is performed, using Eqs. (5)–(9), for $k = 1,2,\ldots$, until the convergence tolerance is satisfied. Let $\lambda_i^{(k+1)}$ be the approximation for λ_i calculated in the kth iteration, $k \geqslant 2$, then we have convergence to an accuracy of $2s$ digits in the required eigenvalues when for $tol = 10^{-2s}$

$$\left[1 - \frac{\left(\lambda_i^{(k+1)}\right)^2}{\left(\mathbf{q}_i^{(k+1)}\right)^T\mathbf{q}_i^{(k+1)}}\right]^{1/2} \leqslant tol; \quad i = 1,\ldots,p \tag{10}$$

where $\mathbf{q}_i^{(k+1)}$ is the i*th* vector in the matrix $\mathbf{Q}_{k+1}$ corresponding to $\lambda_i^{(k+1)}$ [16]. The eigenvector approximations will only be accurate to s digits. The theoretical convergence rates of these vectors are $\frac{\lambda_i}{\lambda_{q+1}}$, with hence a higher convergence rate to an eigenvector corresponding to a smaller eigenvalue [14,16]. While theoretical, these convergence rates are usually also observed in practice.

Third, the Sturm sequence check is carried out to ensure that the lowest p eigenpairs, that is $(\lambda_i, \boldsymbol{\varphi}_i)$ for $i = 1,\ldots p$, have indeed been calculated [1,16]. In case the Sturm sequence check is not passed, it is usually best to continue the iteration with a larger number of iteration vectors. In practice, this condition is hardly encountered provided a large enough number of iteration vectors is employed.

Using the above equations, it is crucial to establish effective starting iteration vectors, considering the *quality* and the *number of vectors*. The *quality* of the starting iteration vectors is important, because theory tells that if the subspace spanned by these vectors contains the exact eigenvectors, then a single iteration will calculate the exact eigenvalues and vectors sought. Nevertheless, in the present paper, we choose to use the simple algorithm of Ref. [1], see also Ref. [16], to establish the starting iteration vectors because we want to focus on other aspects of the solution scheme.

However, it should be noted that starting iteration vectors of much better quality may be generated or known from a previous solution. The eigenvectors just computed can be used, for example, in optimization problems of structures when the frequencies are calculated as the structure changes [16], in solving random eigenvalue problems when using Monte Carlo simulations [13], or in computational biology when evaluating the frequencies and mode shapes of proteins on conformational pathways [19]. In these cases, the use of the calculated eigenvectors of the previous solution as the starting vectors of the next solution can be very effective.

When these conditions do not apply and the order of the matrices n is large, particularly good quality starting iteration vectors may be generated, for example, using a reduction technique or the method of component mode synthesis [16]. Using substructuring smaller systems would be solved, even only approximately, and the solutions of those would be used to establish good starting iteration vectors in $\mathbf{X}_1$ for the complete system solution. This approach can be quite effective if 'typical smaller systems' can be identified which in the complete system repeat themselves, so that the eigensolution of a small problem can be used a multiple number of times in establishing the starting iteration vectors.

Whichever algorithm is used to establish the starting iteration vectors, an effective *number of vectors q* is important because the convergence rate to an eigenvector is given by $\frac{\lambda_i}{\lambda_{q+1}}$. In general, if q is small, but of course larger than p, we need a relatively large number of iterations to converge, while if q is large, we only need a few iterations to converge but in this case each iteration requires more computations. Hence, the use of an effective value of q is desirable and we address the calculation of such value in the next section.