AIAA JOURNAL
Vol. 30, No. 8, August 1992

# Accelerated Subspace Iteration for Eigenvector Derivatives

T. Ting*
*University of Bridgeport, Bridgeport, Connecticut 06601*

**An accelerated subspace iteration method for calculating eigenvector derivatives has been developed. Factors affecting the effectiveness and the reliability of the subspace iteration are identified, and effective strategies concerning these factors are presented. The method has been implemented and the results of a demonstration problem are presented.**

## Introduction

CALCULATION of eigenvector derivatives has become increasingly important in the development of modern numerical methods for areas such as structural optimization, dynamic system identification, and dynamic control. In addition, if the modal expansion technique is employed to predict the dynamic response, it frequently requires the design sensitivity information for a large number of eigenvectors. Thus, an efficient method for calculating derivatives of many eigenvectors is considered herein.

To date, all of the methods for calculating eigenvector derivatives can be categorized into three types: 1) direct method, 2) modal method, and 3) iterative method. Fox and Kapoor[1] presented two methods in 1968 that set solid foundations for the first two methods, i.e., direct and modal methods. Their direct approach was derived based on the first derivative of a single eigenvalue equilibrium equation and its eigenvector mass normalization equation. Thus, this required only the specific eigenvalue and eigenvector of the system that was being considered. Nelson[2] presented a method in 1976 that simplified calculation along this line, and the method was well received. However, since Nelson's method requires solving a set of system-size equations for each eigenvector, it becomes a costly process when the derivatives of a large number of eigenvectors of a large system are demanded. Fox and Kapoor's modal approach employed a modal space expansion concept, wherein the eigenvector derivative spans the entire modal space for the exact solution. They also proposed an approximate solution by spanning a subset of the modal space. The former requires the complete set of eigenpairs, which is prohibitively expensive for large systems. Although the latter only requires a relatively smaller set of eigenpairs, the inaccuracy of the approximation may present a problem. Recently, Wang[3,4] provided two methods to improve this approximation.

This paper is concerned with the third type of method for calculating eigenvector derivatives, i.e., the iterative method. Iterative methods have been proposed by many researchers in this area,[5-7] but most of them suffer from slow convergence rate and, as a result, are less efficient than the direct methods. Lately, Ting[8] derived the basic recursion equations from the subspace iteration equations for solving eigenproblems and improved convergence rate from Bathe's acceleration technique.[9,10] The present work further accelerates the subspace iteration by minimizing the computational cost in each iteration and improving the numerical stability of the iterative process.

## Subspace Iteration Equations

Subspace iteration has been an effective method for solving practical eigenproblems of large $n$ degree-of-freedom systems where only a small number of the lowest $p(p \ll n)$ eigenpairs are required. A typical problem can be stated for a dynamic structural system as

$$\boldsymbol{K}\boldsymbol{\Phi}_p = \boldsymbol{M}\boldsymbol{\Phi}_p\boldsymbol{\Lambda}_p \tag{1}$$

where $\boldsymbol{K}$ and $\boldsymbol{M}$ are the stiffness and mass matrices of the structure, respectively, and $\boldsymbol{\Phi}_p$ and $\boldsymbol{\Lambda}_p$ are the truncated modal and diagonal eigenvalue matrices for the $p$ modes, respectively. In addition, the eigenvectors $\boldsymbol{\Phi}_p$ are usually normalized by the unit generalized mass rule such that

$$\boldsymbol{\Phi}_p^T\boldsymbol{M}\boldsymbol{\Phi}_p = \boldsymbol{I} \tag{2}$$

In general, Eq. (1) can also be expressed in the following so-called subspace iteration equations:

$$(\boldsymbol{K} + \sigma\boldsymbol{M})\boldsymbol{W} = \boldsymbol{M}\boldsymbol{\Phi}_p \tag{3}$$

$$\tilde{\boldsymbol{K}}\boldsymbol{P} = \tilde{\boldsymbol{M}}\boldsymbol{P}\boldsymbol{\Lambda}_p \tag{4}$$

$$\boldsymbol{\Phi}_p = \boldsymbol{W}\boldsymbol{P} \tag{5}$$

where

$$\tilde{\boldsymbol{K}} = \boldsymbol{W}^T\boldsymbol{K}\boldsymbol{W} \tag{6}$$

$$\tilde{\boldsymbol{M}} = \boldsymbol{W}^T\boldsymbol{M}\boldsymbol{W} \tag{7}$$

and for an arbitrary shift value $\sigma$,

$$\boldsymbol{P} = \boldsymbol{\Lambda}_p + \sigma\boldsymbol{I} \tag{8}$$

Applying a shift to Eq. (1) yields

$$(\boldsymbol{K} + \sigma\boldsymbol{M})\boldsymbol{\Phi}_p = \boldsymbol{M}\boldsymbol{\Phi}_p(\boldsymbol{\Lambda}_p + \sigma\boldsymbol{I})$$

Postmultiplying by $(\boldsymbol{\Lambda}_p + \sigma\boldsymbol{I})^{-1}$, the shifted equation becomes

$$(\boldsymbol{K} + \sigma\boldsymbol{M})\boldsymbol{\Phi}_p(\boldsymbol{\Lambda}_p + \sigma\boldsymbol{I})^{-1} = \boldsymbol{M}\boldsymbol{\Phi}_p$$

This demonstrates the derivation of Eqs. (3) and (5). Equation (4) represents a reduced eigenproblem, which is formed by projecting $\boldsymbol{K}$ and $\boldsymbol{M}$ onto a space spanned by columns of $\boldsymbol{W}$. By substituting Eqs. (6-8) into Eq. (4), we can show an equality verification for Eq. (4).

Equations (3-5) combined imply a recurrence relation for the solution of $\boldsymbol{\Phi}_p$ and $\boldsymbol{\Lambda}_p$. In addition to progressively solving

Presented as Paper 91-1055 at the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, Baltimore, MD, April 8-10, 1991; received June 28, 1991; revision received Jan. 27, 1992; accepted for publication Jan. 28, 1992. Copyright © 1991 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Associate Professor, Mechanical Engineering Department. Member AIAA.

this equation and iterating until it converges, the process only requires a set of initial trial vectors for $\Phi_p$ in Eq. 3.

## Eigenvector Derivatives

Three methods for computing the derivatives of $\Phi_p$ with respect to design variables $x_j$, $j = 1, 2, \ldots, m$ are to be described here. These methods represent three distinct approaches to solving the problem.

### Nelson's Method

Nelson's method deals with one eigenmode at a time. Thus, consider the eigenproblem equation for the $i$th mode:

$$(\boldsymbol{K} - \lambda_i \boldsymbol{M})\phi_i = 0 \tag{9}$$

The first partial derivative of Eq. (1) with respect of $x_j$ gives

$$(\boldsymbol{K} - \lambda_i \boldsymbol{M})\phi_{i,j} = -\boldsymbol{K}_{,j}\phi_i + \lambda_{i,j}\boldsymbol{M}\phi_i + \lambda_i \boldsymbol{M}_{,j}\phi_i \tag{10}$$

or simply

$$\boldsymbol{D}_i \phi_{i,j} = \boldsymbol{f}_i \tag{11}$$

where the subscript $,j$ represents the first derivative with respect to a typical design variable $x_j$.

If $\lambda_i$ is not repeated, the matrix $\boldsymbol{D}_i$ is singular of one rank less and $\phi_{i,j}$ cannot be solved directly from Eq. (11). Instead, we shall first solve a reduced equation of

$$\bar{\boldsymbol{D}}_i \bar{v}_i = \bar{\boldsymbol{f}}_i \tag{12}$$

for $\bar{v}_i$. The reduction of the equation is simply done by eliminating the row and column corresponding to a relatively large component of $\phi_i$. Then let $v_i$ be the expanded $\bar{v}_i$ with a zero inserted at the eliminated position. Now, we state

$$\phi_{i,j} = v_i + c_i \phi_i \tag{13}$$

where $c_i$ is computed based on the first derivative of the mass normalization equation and the modal summation equation for $\phi_{i,j}$, which gives

$$c_i = -\tfrac{1}{2}\phi_i^T \boldsymbol{M}_{,j}\phi_i - \phi_i^T \boldsymbol{M} v_i \tag{14}$$

## Modal Method

Fox and Kapoor's modal method for calculating the derivatives of $\phi_i$ with respect to $x_j$ can be expressed as

$$\phi_{i,j} = \boldsymbol{\Phi} \boldsymbol{a}^{ij} \tag{15}$$

where $\boldsymbol{\Phi}$ is the full modal matrix, and the components of the vector $\boldsymbol{a}^{ij}$ are the modal participation factors for $\phi_{i,j}$. The components of $\boldsymbol{a}^{ij}$ are obtained as the following:

$$a_k^{ij} = \frac{\phi_i^T(\boldsymbol{K}_{,j} - \lambda_i \boldsymbol{M}_{,j})\phi_i}{\lambda_i - \lambda_k} \qquad (\text{for } k \neq i) \tag{16a}$$

$$a_i^{ij} = \frac{\phi_i^T \boldsymbol{M}_{,j}\phi_i}{2} \tag{16b}$$

If there are only $q$ lowest eigenvectors available, $\bar{\phi}_{i,j}$ may be used to approximate $\phi_{i,j}$ where

$$\bar{\phi}_{i,j} = \boldsymbol{\Phi}_q \boldsymbol{a}_q^{ij} \tag{17}$$

$\boldsymbol{\Phi}_q$ and $\boldsymbol{a}_q^{ij}$ are the truncated modal matrix and modal participation vector for the $q$ available eigenvectors, respectively. However, the exact solution $\phi_{i,j}$ can be written as

$$\begin{aligned} \phi_{i,j} &= \boldsymbol{\Phi}_q \boldsymbol{a}_q^{ij} + \boldsymbol{\Phi}_r \boldsymbol{a}_r^{ij} \\ &= \bar{\phi}_{i,j} + \Delta\phi_{i,j} \end{aligned} \tag{18}$$

where $\Delta\phi_{i,j}$ represents the error involved with the approximate solution using incomplete modes.

Wang presented two methods to improve the approximation to $\phi_{i,j}$ with a limited number of available modes. His explicit method simply approximates $\Delta\phi_{i,j}$ by a static mode defined by

$$\boldsymbol{w}_i = \boldsymbol{K}^{-1}\boldsymbol{f}_i - \boldsymbol{\Phi}_q \boldsymbol{\Lambda}_q^{-1}\boldsymbol{\Phi}_q^T \boldsymbol{f}_i \tag{19}$$

In general, any approach to approximate $\Delta\phi_{i,j}$ as a linear combination of $\boldsymbol{\Phi}_r$ results in

$$\tilde{\phi}_{i,j} = \boldsymbol{\Phi}_q \boldsymbol{a}_q^{ij} + \boldsymbol{\Phi}_r \tilde{\boldsymbol{a}}_r^{ij} \tag{20}$$

where $\tilde{\boldsymbol{a}}_r^{ij}$ approximates $\boldsymbol{a}_r^{ij}$.

Wang's implicit formulation further improves the approximation by a Ritz minimization using $\boldsymbol{w}_i$ as an additional assumed mode, i.e.,

$$\begin{aligned} \tilde{\phi}_{i,j} &= [\boldsymbol{\Phi}_q \vdots \boldsymbol{w}_i]\, \boldsymbol{q} \\ &= \boldsymbol{T}_i \boldsymbol{q}_i \end{aligned} \tag{21}$$

Hence, we obtain

$$\boldsymbol{q}_i = (\boldsymbol{T}_i^T \boldsymbol{D}_i \boldsymbol{T}_i)^{-1} \boldsymbol{T}_i^T \boldsymbol{f}_i \tag{22}$$

### Subspace Iteration

The following presents a basic subspace iteration scheme for calculating eigenvector derivatives. The algorithm simultaneously computes derivatives for all of the eigenvectors $\Phi_p$. It begins with a set of trial vectors for $\Phi_{p,j}$ and updates the trial vectors based on the recurrence relations to improve the approximation.

Let $\Psi(0)$ be the set of initial trial vectors. For $k = 0,1,2,\ldots$, the recurrence relations begin with

$$(\boldsymbol{K} + \sigma \boldsymbol{M})\boldsymbol{V}^{(k+1)} = \bar{\boldsymbol{F}} + \boldsymbol{M}\boldsymbol{\Psi}^{(k)} \tag{23}$$

where

$$\bar{\boldsymbol{F}} = \boldsymbol{M}_{,j}\boldsymbol{\Phi}_p - (\boldsymbol{K}_{,j} + \sigma \boldsymbol{M}_{,j})\boldsymbol{W} \tag{24}$$

is constant throughout the iterations. Solve for $\boldsymbol{V}^{(k+1)}$ in Eq. (23) and evaluate

$$\tilde{\boldsymbol{K}}_{,j}^{(k+1)} = \boldsymbol{W}^T \boldsymbol{K}_{,j} \boldsymbol{W} + \boldsymbol{W}^T \boldsymbol{K} \boldsymbol{V}^{(k+1)} + (\boldsymbol{W}^T \boldsymbol{K} \boldsymbol{V}^{(k+1)})^T \tag{25}$$

$$\tilde{\boldsymbol{M}}_{,j}^{(k+1)} = \boldsymbol{W}^T \boldsymbol{M}_{,j} \boldsymbol{W} + \boldsymbol{W}^T \boldsymbol{M} \boldsymbol{V}^{(k+1)} + (\boldsymbol{W}^T \boldsymbol{M} \boldsymbol{V}^{(k+1)})^T \tag{26}$$

Next, consider the following relation:

$$\tilde{\boldsymbol{K}}\boldsymbol{D}^{(k+1)} - \tilde{\boldsymbol{M}}\boldsymbol{D}^{(k+1)}\boldsymbol{\Lambda}_p = \tilde{\boldsymbol{M}}\boldsymbol{P}\boldsymbol{\Lambda}_{p,j} + \tilde{\boldsymbol{M}}_{,j}^{(k+1)}\boldsymbol{P}\boldsymbol{\Lambda}_p - \tilde{\boldsymbol{K}}_{,j}^{(k+1)}\boldsymbol{P} \tag{27}$$

and solve for $\boldsymbol{D}^{(k+1)}$ using a Nelson-like method. Then, for the next iteration, we shall use the updated trial vectors

$$\boldsymbol{\Psi}^{(k+1)} = \boldsymbol{V}^{(k+1)}\boldsymbol{P} + \boldsymbol{W}\boldsymbol{D}^{(k+1)} \tag{28}$$

The iteration process converges, so that

$$\boldsymbol{\Psi}^{(k+1)} \to \boldsymbol{\Phi}_{p,j} \qquad \text{as} \qquad k \to \infty \tag{29}$$

## Acceleration

The computational efficiency of an iterative scheme depends on three major factors. These are 1) the overall convergence rate, 2) computation cost in each iteration, and 3) numerical stability. To accelerate the subspace iteration for calculating eigenvector derivatives, a thorough study was conducted of all of the possibilities in improving the numerical conditions with regard to these three factors.

In each iteration of the basic scheme, computation is carried out consecutively through three equations, i.e., Eqs. (23), (27), and (28). Basically, Eq. (23) involves a forward/backward substitution to generate a set of $q$ new iteration vectors for derivatives of $p$ eigenvectors with respect to each design variable. Next, a Ritz minimization is performed in Eq. (27) to determine $\boldsymbol{D}^{(k+1)}$. Solutions of Eqs. (23) and (27) are then used in Eq. (28) to update the trial vectors for the eigenvector derivatives.

In the iteration process, Eq. (23) controls the overall convergence rate and is the key step in improving the approximation at each iteration. On the other hand, it is also a major source of errors that may propagate to cause a divergence of the process. With this concern, Eq. (27) not only determines $\boldsymbol{D}^{(k+1)}$, it also serves to correct errors occurring in Eq. (23). However, the computation cost involved in solving Eq. (27) is considerably high, which greatly influences the overall effectiveness of the method. Therefore, it becomes a necessity to reduce or even to eliminate the need to actually perform the solution for Eq. (27).

#### Convergence Rate of Subspace Iteration

The ultimate rate of convergence for subspace iteration is the same for both the eigenproblem and the eigenvector derivatives. This is due to the fact that the left-hand-side matrix $(\boldsymbol{K} + \sigma\boldsymbol{M})$ is the same in both iteration equations (3) and (23). That is, referring to Bathe's work, for each iteration vector associated with the $i$th mode, the ultimate convergence rate is $(\lambda_i + \sigma)/(\lambda_{q+1} + \sigma)$ where $q$ is the number of the eigenvectors spanning the subspace. A large $q$ $(q > p)$ is normally required in order to achieve a good convergence rate.

#### Overrelaxation

Overrelaxation techniques have been used to accelerate the convergence rate of many slowly converging iterative procedures. The overall convergence rate of the subspace iteration is mostly determined by the solution characteristics of Eq. (23). Comparing Eq. (23) to the original subspace iteration equation for solving the eigenproblem, Eq. (3), there are two points of similarity that can be indicated. Both equations have the same left-hand-side matrix to be either inverted or decomposed for solving the unknown vectors. On the right-hand side, the original subspace iteration normally will converge to a vector subspace and the other utilizes this resulting subspace for forming trial vectors. These similarities suggest that we may employ any existing overrelaxation scheme designed for the original subspace iteration to accelerate the present iteration process.

Considering Bathe's overrelaxation method, the present subspace iteration for eigenvector derivatives can be overrelaxed by letting

$$\boldsymbol{\Psi}^{(k+1)} = \boldsymbol{\Psi}^{(k)} + [(\boldsymbol{V}^{(k+1)}\boldsymbol{P} + \boldsymbol{W}\boldsymbol{D}^{(k+1)}) - \boldsymbol{\Psi}^{(k)}]\boldsymbol{\alpha} \tag{30}$$

where $\alpha$ is a diagonal matrix with its diagonal terms equal to individual vector overrelaxation factors $\alpha_i$, $i = 1, \ldots, q$. Referring to Bathe's analysis,[6] we have

$$\alpha_i = \frac{1}{1 - (\lambda_i + \sigma)/(\lambda_{q+1} + \sigma)} \tag{31}$$

which overrelaxes iteration vectors corresponding to the derivatives of $\phi_i$. It should be noted that the advantage we have is that all of the eigenvalues used to evaluate $\alpha$ are known a priori.

#### Trial Vectors

In general, the subspace iteration theory indicates that a large subspace may result in a rapid convergence. Thus, in the basic subspace iteration for eigenvector derivatives, we need to include all $q$ available eigenvectors to improve the convergence rate in computing derivatives of $p$ $(q > p)$ eigenvectors. But the extra effort required to carry the extra $(q\text{-}p)$ iteration vectors for every design variable in the iteration process may well offset the savings gained in speeding up the convergence. However, it can be shown that regardless of what $\boldsymbol{\Psi}^{(k)}$ are being used as the trial vectors in Eq. (23), the basic scheme produces a set of new trial vectors $\boldsymbol{\Psi}^{(k+1)}$, which can always be represented by Eq. (20). This indicates that Eq. (27) restricts $\boldsymbol{\Psi}^{(k+1)}$ to being in the solution space expressed by Eq. (20). Further, it can also be shown that any set of trial vectors in the space of Eq. (20) will result in a diagonal $\boldsymbol{D}^{(k+1)}$ and

$$\boldsymbol{D}^{(k+1)} = \boldsymbol{\Lambda}_{q,j} \tag{32}$$

where $\boldsymbol{\Lambda}_{q,j}$ is diagonal and can be determined analytically. As a result, Eq. (28) becomes an uncoupled equation for updating the trial vectors, and all of the iteration vectors are independent of each other in the entire process.

Consequently, if we use the truncated modal approximation of Eq. (17) to construct the initial trial vectors using all of the $q$ available eigenvectors, we will gain many computational advantages. First of all, only the necessary trial vectors are required in the iterations, and the same convergence rate as with all $q$ vectors will be maintained. Secondly, since $\boldsymbol{D}^{(k+1)}$ becomes constant and can be predetermined, the exhaustive computation required for Eq. (27) can be eliminated. Instead, we can always assign $\boldsymbol{D}^{(k+1)}$ by

$$\boldsymbol{D}^{(k+1)} = \boldsymbol{\Lambda}_{p,j} \tag{33}$$

#### Numerical Stability

As mentioned earlier, Eq. (23) is responsible for both the convergence rate and the numerical stability of the entire iteration process. The numerical stability problem is normally attributed to the accumulation of roundoff errors occurring in the forward/backward substitution of Eq. (23). In the basic scheme, this problem is automatically corrected by a rigorous evaluation of $\boldsymbol{D}^{(k+1)}$, using Eq. (27). However, if Eq. (27) is eliminated in the computation, the roundoff errors accumulate in the iterations and may lead to a divergence of the process.

Now we shall examine the solution of Eq. (23) and investigate the possible source of errors resided in the solution. Consider a typical solution vector $\boldsymbol{v}^{(k+1)}$ of Eq. (23) and express it in the form of modal summation, i.e.,

$$\boldsymbol{v}^{(k+1)} = \sum_{k=1}^{n} b_k \phi_k \tag{34}$$

where $b_k$ are the modal participation factors for $\boldsymbol{v}^{(k+1)}$.

It can abe demonstrated, using a modal decomposition technique, that each $b_k$ is a function of $1/(\lambda_k + \sigma)$. This relation indicates that greater errors are more likely to appear in the first few terms of Eq. (34) than in the rest of the terms. This means that the errors are most likely to come from the participating lower modes. Fortunately, the subspace employed here is usually spanned by the known $q$ lowest eigenvectors, and they can provide an economical remedy to this problem.

It was indicated earlier that an updated trial vector $\psi_{ij}^{(k+1)}$ for $\phi_{i,j}$ can be defined as in Eq. (20), i.e.,

$$\psi_{ij}^{(k+1)} = \boldsymbol{\Phi}_q \boldsymbol{a}_q^{ij} + \boldsymbol{\Phi}_r \tilde{\boldsymbol{a}}_r^{ij} \tag{35}$$

where $\boldsymbol{\Phi}_q$ are the known $q$ eigenvectors of the system and the components of $\boldsymbol{a}_q^{ij}$ are determined by Eqs. (16). Suppose the $\psi_{ij}^{(k+1)}$ is contaminated by the participating lower modes and results in

$$\hat{\psi}_{ij}^{(k+1)} = \boldsymbol{\Phi}_q \hat{\boldsymbol{a}}_q^{ij} + \boldsymbol{\Phi}_r \tilde{\boldsymbol{a}}_r^{ij} \tag{36}$$

Employing the modal orthonormal property, we have

$$\hat{\boldsymbol{a}}_q^{ij} = \boldsymbol{\Phi}_q^T \boldsymbol{M} \hat{\boldsymbol{\Psi}}_{ij}^{(k+1)} \tag{37}$$

**Table 1 Efficiency comparison for six modes**

| | Basic subspace | | Accelerated | Nelson's |
|---|---|---|---|---|
| Number of iteration vectors | 6 | 12 | 6 | — |
| Number of iterations | 12 | 5 | 4 | — |
| CPU seconds | 539.7 | 494 | 244.5 | 667.5 |

**Table 2 Efficiency comparison for twelve modes**

| | Basic subspace | | Accelerated | Nelson's |
|---|---|---|---|---|
| Number of iteration vectors | 12 | 20 | 12 | — |
| Number of iterations | 16 | 7 | 6 | — |
| CPU seconds | 1181.7 | 1004 | 492.9 | 1293.6 |

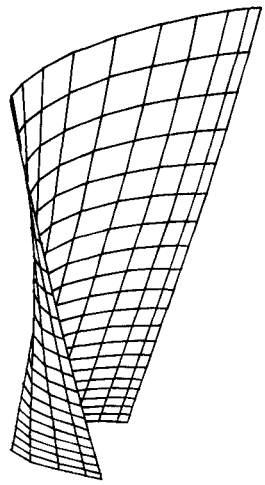

**Fig. 1 Power turbine blade finite element model.**

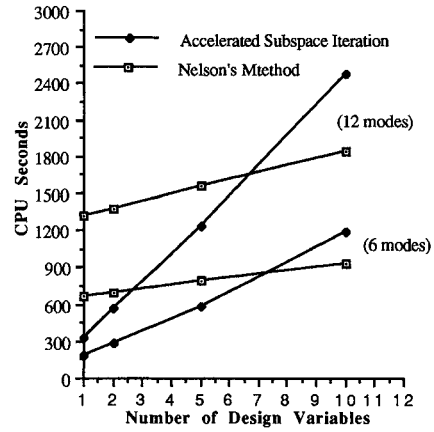


**Fig. 2 CPU time vs number of design variables.**

Thus, a subspace model correction can be given as

$$\psi_{ij}^{(k+1)} = \hat{\psi}_{ij}^{(k+1)} + \Phi_q(a_q^{ij} - \hat{a}_q^{ij}) \tag{38}$$

**Convergence Check**

The convergence check should be performed virtually in every iteration. For a large number of iteration vectors, the computation involved in a convergence check is desirable to be minimized.

Applying the fixed-point theorem, let us define a vector change measure for each iteration vector by

$$\epsilon_{ij} = \frac{|\psi_{ij}^{(k+1)} - \psi_{ij}^{(k)}|}{|\psi_{ij}^{(k+1)}|} \tag{39}$$

where $|\ |$ denotes a vector norm of any kind. The most commonly used vectors are the $L_p$ norms. For instance, $L_2$ norm is the Euclidean norm and $L_\infty$ norm is the maximum norm. In most cases, the maximum norm requires the least computation.

An iteration vector converges when

$$\epsilon_{ij} \leq \text{tol} \tag{40}$$

where tol is the convergence tolerance and is typically $10^{-3}$ to $10^{-5}$.

Since the accelerated subspace iteration is completely uncoupled, any converged vectors may be partitioned out and stored at any time to save the computation. However, a careful bookkeeping should be maintained for the final assemblage.

## Accelerated Algorithm

A computer algorithm to accelerate the basic subspace iteration for eigenvector derivatives has been implemented in MSC/NASTRAN. It integrates all of the techniques discussed in the previous section to increase the effectiveness as well as to maintain the reliability of the iterative scheme.

All of the data, including the structural matrices and eigensolution, should be fetched from the preceding analysis data base. It is assumed that the decomposition of $(\boldsymbol{K} + \sigma\boldsymbol{M})$ was performed in the analysis phase for a suitable shift value of $\sigma$ and the corresponding decomposition factors were saved. For example, a decomposition of

$$\boldsymbol{K} + \sigma\boldsymbol{M} = \boldsymbol{L}^T\boldsymbol{L} \tag{41}$$

results in a lower factor $\boldsymbol{L}$, which should be saved along with the shift value $\sigma$ in the preceding eigenvalue analysis run.

The algorithm should begin by fetching all of the necessary data from the data base generated in the preceding analysis phase. dIt is then followed by the calculation of eigenvalue derivatives. Before the subspace iterations take place, it is important to generate a set of good trial vectors. Based on the preceding discussion, Eq. (17) should be used to generate these initial vectors. The subspace iterations can then begin, and the iterations will continue until all of the vectors have converged.

Typically, the $(k+1)$th iteration takes the following steps:

1) Perform forward/backward substitutions to solve for $V^{(k+1)}$ [Eq. (23)].

2) Define $D^{(k+1)}$ based on Eq. (33).

3) Perform Bathe's vector overrelaxation [Eq. (30)].

4) Update the trial vectors [Eq. (28)].

5) Perform subspace modal correction to trial vectors [Eq. (38)].

6) Conduct a convergence check [Eq. (40)] and partition out the converged vectors.

Steps 3 and 5 should not be conducted every iteration. They need be done only once every several, say three, iterations. In fact, step 3 cannot be conducted more than once in every three iterations in order to provide sufficient time for the numerical overrelaxation to settle down and to prevent premature divergence of the process. Step 5 is required periodically, however, to prevent the roundoff errors from overaccumulating over many iterations.

## Example Problem

The finite element model of a power turbine blade with a cantilevered end condition is used to demonstrate the accelerated subspace iterations for calculating eigenvector derivatives. The model consists of 200 flat-plate elements comprising 231 grid points, as shown in Fig. 1. The elements are divided into 20 rows, from the root to the tip, of 10 elements each.

The design variables are defined as scale factors of element thicknesses, and each design variable links to the thicknesses of all of the elements in a row. The efficiency of the subspace iteration method for computing the eigenvector derivatives with respect to the design variable has been compared with that of Nelson's method. Both Nelson's[11] and the subspace iteration methods have been implemented in MSC/NASTRAN's DMAP language.[12]

Tables 1 and 2 summarize the comparisons of CPU time for calculating derivatives of 6 and 12 eigenvectors, respectively, with respect to one design variable. With the basic scheme of subspace iteration, results for two different numbers of iteration vectors are presented. It is shown that using the minimum number of vectors resulted in a much slower converging process when compared with using the proper number of vectors as suggested by Bathe for subspace iteration. Overall, subspace iteration methods are more efficient than Nelson's method in this one design variable case, and the accelerated scheme increases the efficiency significantly from the basic scheme. However, when the number of design variables increases, the accelerated subspace iteration gradually losses its superiority over Nelson's method, as depicted in Fig. 2.

This example was run on a DEC VAX-11/785 and the CPU seconds shown are the standard VAX CPU seconds. A convergence check was performed in each subspace iteration where the change measures were determined according to Eq. (41) for all of the iteration vectors. The maximum vector norm, i.e., $L_\infty$ norm, was used for the accelerated scheme in the determination of the iteration vector change measures, and the Euclidean vector norm was used for the basic scheme. The tolerance for $\epsilon_i$ was set to be $5 \times 10^{-4}$ in both cases.

## Conclusions

This paper analyzes the factors influencing the efficiency of the subspace iteration method for computing eigenvector derivatives of real and symmetric matrices and provides effective strategies to accelerate the basic subspace iteration for all possibilities considered here. As a result, an accelerated subspace iteration algorithm is presented as an integration of all of these procedures. The accelerated algorithm has been implemented, and the results of a sample problem are reported.

The accelerated scheme retains the advantageous features of the basic scheme such as the elimination of the need to solve a large system of equations for each interested mode and the optimal use of all available results obtained in the preceding eigenanalysis. A complete uncoupling of all iteration vectors in the accelerated scheme is a major improvement over the basic subspace iteration. This not only simplifies the computation in each iteration, it also allows the convergence of each vector to be controlled individually. The use of Bathe's overrelaxation technique provides an acceleration on the overall convergence speed. Finally, an economical subspace modal correction is introduced to maintain the numerical stability of iteration vectors.

More than 70% of the savings in CPU time over Nelson's method was observed when using the present method to compute the derivatives of many eigenvectors with respect to a single design variable, as demonstrated in the example problem. On the other hand, as the number of design variables increases, the efficiency superiority of the present method decreases or even reverses itself. However, the system size also plays a decisive factor in the overall effectiveness of the two methods. Nevertheless, the accelerated subspace iteration definitely wins its place in the areas of applications where a small number of design variables are involved. A typical example is the calculation of directional derivatives during the line search of an optimization process.

## Acknowledgments

The author gratefully acknowledges the partial support of this effort from NASA Langley Research Center under NASA Grant NAG-1-1033 and Sikorsky Aircraft Corporation under Contract S2530989.

## References

[1]Fox, R. L., and Kapoor, M. P., "Rates of Change of Eigenvalues and Eigenvectors," *AIAA Journal*, Vol. 6, No. 12, 1968, pp. 2426–2429.

[2]Nelson, R. B., "Simplified Calculation of Eigenvector Derivatives," *AIAA Journal*, Vol. 1, No. 9, 1976, pp. 1201–1205.

[3]Wang, B. P., "An Improved Approximate Method for Computing Eigenvector Derivatives," AIAA/ASME/ASCE/AHS 26th Structures, Structural Dynamics and Materials Conf., Orlando, FL, April 1985.

[4]Wang, B. P., "Improved Approximate Methods for Computing Eigenvector Derivatives in Structural Dynamics," *AIAA Journal*, Vol. 29, No. 6, pp. 1018–1020.

[5]Rusidill, C. S., and Chu, Y., "Numerical Methods for Evaluating the Derivatives of Eigenvalues and Eigenvectors," *AIAA Journal*, Vol. 13, No. 6, 1975, pp. 834–837.

[6]Andrew, A. L., "Convergence of an Iterative Method for Derivatives of Eigensystems," *Journal of Computational Physics*, Vol. 26, 1978, pp. 107–112.

[7]High, G. D., "An Iterative Method for Eigenvector Derivatives," *MSC World Users Conference Proceedings*, Vol. 1, MacNeal-Schwendler Corp., Los Angeles, CA, March 1990.

[8]Ting, T., "An Efficient Design Sensitivity Analysis of Eigenvectors," 3rd Air Force/NASA Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, San Francisco, CA, Sept. 1990, pp. 507–598.

[9]Bathe, K. J., "Convergence of Subspace Iteration," *Formulations and Numerical Algorithms in Finite Element Analysis*, edited by K. J. Bathe, J. J. Oden, and W. Wunderich, MIT Press, Cambridge, MA, 1977, Chap. 19.

[10]Bathe, K. J., and Ramaswamy, S., "An Accelerated Subspace Iteration Method," *Computer Method in Applied Mechanics and Engineering*, Vol. 23, pp. 313–331.

[11]Ting, T., and Wallerstein, D., "Calculation of Eigenvector Derivatives in Design Sensitivity Analysis," *MSC/NASTRAN Application Manual*, Application Note, MacNeal Schwendler, Los Angeles, CA, 1986.

[12]Gockel, M. A., et al (eds.), *MSC/NASTRAN User's Manual, II*, MacNeal Schwendler, Los Angeles, CA, 1985.