AIAA JOURNAL
Vol. 35, No. 11, November 1997

Efficient Computation of Eigenvector Sensitivities for Structural Dynamics

Kenneth F. Alvin*
Sandia National Laboratories, Albuquerque, New Mexico 87185

An iterative procedure is presented for computing eigenvector sensitivities due to finite element model parameter variations. The present method is a preconditioned conjugate projected gradient-based technique and is intended to utilize the existing matrix factorizations developed for an iterative eigensolution such as Lanczos or Subspace Iteration. As such, this technique can be integrated into a coupled eigensolver/sensitivity software module and leverage the nonrecurring costs of the solver. The use of projection operators in the algorithm is dictated by the indefinite character of the governing coefficient matrix of the eigenvector derivative. Two model examples are provided to demonstrate both the accuracy of the present technique and its superior efficiency as compared to existing techniques with similar accuracy.

I. Introduction

THE computation of modal parameter sensitivities is a necessary component of design optimization, uncertainty analysis, and the validation of structural dynamics models. These sensitivities represent a linearized estimate of the change in the modal parameters, principally frequencies and mode shapes, due to perturbations of the stiffness and mass matrices of the model. Many methods have been proposed for accurate and efficient computation of these sensitivity coefficients.[1-5] These methods compute eigenvalue sensitivities in the same way but differ in their techniques for eigenvector sensitivities. The present work presents an improvement of the existing methods for problems with distinct roots; the problem of repeated roots[6-8] is not addressed in this study.

Computationally, the simplest method for approximating the derivatives of eigenvectors is by modal superposition, in which the desired quantities are assumed to be adequately represented by contributions of known eigenvectors.[1] This approach, often referred to as Fox's method, has negligible computational cost but can be inaccurate if an incomplete set of eigenvectors is used in the expansion (as would normally be the case for practical finite element models). To circumvent these problems, a number of alternative formulations have been proposed, including static corrections[3] and iteration.[4] In contrast, the direct exact method, termed Nelson's method,[2] requires a costly factorization of the dynamic stiffness matrix $(K - \lambda_i M)$ for each mode i. Because this dynamic stiffness matrix is indefinite and singular, the matrix factorization suggested by Nelson requires pivoting, which is expensive in terms of CPU and memory.

The technique presented here builds on some of these ideas, while also seeking to utilize the previously computed and factorized shifted stiffness matrix from the computation of the eigenvalues and eigenvectors themselves. The result is an iterative method based on the preconditioned conjugate gradient (PCG) algorithm, where the desired solution is added to the modal superposition solution to obtain accuracy similar to Nelson's method. The indefinite character of the dynamic stiffness matrix $(K - \lambda_i M)$ requires careful projections of the residual and solution vectors to ensure convergence of conjugate gradients (CGs), leading to a preconditioned conjugate projected gradient (PCPG) algorithm. The preconditioner is the shifted stiffness matrix $(K - \sigma M)$, where $\sigma \leq 0$ typically for structures with low frequencies or rigid-body modes. This shifted stiffness matrix is the same operator that is repeatedly solved in iterated eigensolution algorithms such as Lanczos or Subspace Iteration. Thus, by developing a coupled eigensolver/sensitivity software module, the same solver can be utilized for computing the eigenvector derivatives. Furthermore, if a direct solver is employed for $(K - \sigma M)$, the existing factorization can be retained in memory to be used for the sensitivity computations, saving a portion of the total computational cost.

Beyond generating improved estimates of the modal vector derivatives, the present procedure was motivated by the future need of a massively parallel finite element analysis package for structural dynamics that includes both modal solutions and sensitivity analysis capabilities to support model validation and nondeterministic analysis. Such a code, developed for a distributed memory multiprocessor environment, will be built around an scalable iterative solver. This iterative solver shares with direct solvers the property that the average cost per solve decreases quickly as the number of solves increase. The present technique can leverage off nearly all of the nonrecurring costs of this solver if it is executed immediately following the convergence of the nominal eigensolution.

The present iterative approach for determining eigenvector derivatives, using the modal superposition solution as its initial point, allows the user a great deal of flexibility in adjusting the algorithm to the available time and memory resources. For example, if many more eigenmodes are computed than modes requiring sensitivities, the modal superposition solution may be adequate or at least accelerate the convergence of the sensitivities. On the other hand, if it would be expensive to extract additional modes for the sensitivity calculation, then it may be more efficient to perform the iterative sensitivity computation directly with the existing set of eigenmodes. The advantage here is that the number of modes required for an accurate modal superposition solution is never clear a priori and can only be assessed by computing the residual on which the present technique is based.

The remainder of the paper is organized as follows. In Sec. II, a preliminary review of the theoretical background is provided. Section III details some of the existing methods for computing eigenvector derivatives, including modal superposition and Nelson's method. In Sec. IV, the new PCPG-based algorithm is developed and summarized. Section V provides accuracy and performance comparisons between existing methods and the present procedure. Finally, concluding remarks are offered in Sec. VI.

II. Background

The governing undamped eigenproblem for structural dynamics is written as

$$\begin{aligned} K\phi_i &= \lambda_i M\phi_i \\ \phi_i^T M\phi_j &= \delta_{ij} \end{aligned} \qquad i = 1, \ldots, n \tag{1}$$

where K and M are the stiffness and mass matrices, respectively; λ_i is the ith eigenvalue, equal to ω_{ni}^2, the square of the natural frequency

Received Feb. 1, 1997; revision received July 9, 1997; accepted for publication July 15, 1997.

*Senior Member, Technical Staff, Structural Dynamics and Vibration Control Dept., P.O. Box 5800. E-mail: kfalvin@sandia.gov. Member AIAA.

in radian per second; ϕ_i is the associated eigenvector; and δ_{ij} is the Kronecker delta. To obtain the derivatives of the modal parameters in terms of the derivatives of the stiffness and mass, we differentiate Eq. (1) to obtain

$$\begin{aligned}(K-\lambda_i M)\phi_i' + (K' - \lambda_i M' - \lambda_i' M)\phi_i = 0\\ \phi_i^T M'\phi_i + 2\phi_i^T M\phi_i' = 0\end{aligned} \tag{2}$$

where $(\,)'$ is the derivative of the given quantity with respect to some model parameter. For systems with distinct eigenvalues, i.e., no repeated roots, Eqs. (1) and (2) are sufficient to uniquely determine the rates of change of the eigenvalue λ_i and the eigenvector ϕ_i.

A. Eigenvalue Sensitivity

To determine the eigenvalue derivative λ_i', we multiply Eq. (2) by ϕ_i^T and use Eq. (1), viz.,

$$\begin{aligned}\phi_i^T(K' - \lambda_i M' - \lambda_i' M)\phi_i = 0\\ \therefore \lambda_i' = \phi_i^T(K' - \lambda_i M')\phi_i\end{aligned} \tag{3}$$

B. Eigenvector Sensitivity

Given that the eigenvalue sensitivity has been determined, Eq. (2) can be written as

$$(K-\lambda_i M)\phi_i' = \boldsymbol{f}_i \tag{4}$$

where

$$\boldsymbol{f}_i = -(K' - \lambda_i M' - \lambda_i' M)\phi_i \tag{5}$$

Because $K - \lambda_i M$ is singular, Eq. (4) does not determine the component of ϕ_i' along ϕ_i. Therefore, differentiate $\phi_i^T M\phi_i$ to obtain

$$\phi_i^T M\phi_i' = -\tfrac{1}{2}\phi_i^T M'\phi_i \tag{6}$$

Thus, if λ_i is a simple eigenvalue,

$$\phi_i' = \left[-\tfrac{1}{2}\phi_i^T M'\phi_i - \phi_i^T M(K-\lambda_i M)^+ \boldsymbol{f}_i\right]\phi_i + (K-\lambda_i M)^+ \boldsymbol{f}_i \tag{7}$$

III. Previous Research and Methods

Many researchers have studied methods for determining modal parameter sensitivities, primarily focusing on the problem of determining eigenvector derivatives and on the complications posed by repeated modes. In the present effort, we are interested in efficiently and accurately determining eigenvector derivatives, and so we will assume that the eigenvectors are distinct, i.e., simple. A brief description of some of the most important existing methods follows.

A. Nelson's Exact Method

Nelson's method[2] is to first solve the equation $(K-\lambda_i M)\boldsymbol{z}_i = \boldsymbol{f}_i$ by a generalized inverse of the singular operator and then compute the eigenvector derivative, viz.,

$$\phi_i' = \left(-\tfrac{1}{2}\phi_i^T M'\phi_i - \phi_i^T M\boldsymbol{z}_i\right)\phi_i + \boldsymbol{z}_i \tag{8}$$

Here $\boldsymbol{z}_i$ is a pseudoinverse solution computed by setting the jth row and column of $(K-\lambda_i M)$ and the jth element of $\boldsymbol{f}_i$ equal to zero, while the jth diagonal element of $(K-\lambda_i M)$ is set equal to 1.0. The row-column index j is usually chosen to correspond to the largest magnitude element in the eigenvector ϕ_i.

The distinguishing characteristic of Nelson's method is that it is computationally intensive but gives the exact solution for the derivative. Therefore, Nelson's method is often employed as a benchmark in assessing the accuracy of approximate solutions. The computational problem posed by Nelson's method is that it requires the factorization of an indefinite matrix for each mode considered. Thus, for even a small number of modes and design variables, the cost of implementing Nelson's method can easily dwarf the cost of solving the nominal eigenproblem.

B. Fox's Modal Superposition Method

Because Nelson's method is impractical in many situations involving large-order problems, a simple alternative is to assume that the desired eigenvector derivative can be written as the superposition of the mode shapes already computed and available from the nominal eigenproblem. The modal superposition solution,[1] also known as Fox's method, is obtained by defining the eigenvector derivative as

$$\phi_i' = \Phi\boldsymbol{c}_i \tag{9}$$

where Φ is the set of computed or known eigenvectors and $\boldsymbol{c}_i$ is a vector to be determined. Substituting into Eq. (4) and premultiplying by ϕ_j^T $(j \neq i)$, we have

$$\boldsymbol{c}_i(j) = \begin{cases}\dfrac{(\phi_j^T \boldsymbol{f}_i)}{(\lambda_j - \lambda_i)} & j \neq i\\ -\tfrac{1}{2}\phi_i^T M'\phi_i & j = i\end{cases} \tag{10}$$

By inspection, Fox's method requires very little computation, but it has frequently been shown to be highly inaccurate. Although the accuracy problems can be mitigated to some extent by increasing the number of modes computed in the nominal eigenproblem, this also increases the computational effort and the cutoff for determining how many modes are sufficient is very unclear. For this reason, Wang[3] proposed adding a single static mode correction to Fox's method to approximate the effect of the omitted modes.

C. Wang's Method[3]

An extension of the modal superposition method can be obtained by adding a residual static mode correction as follows. Write the eigenvector derivative as

$$\phi_i' \approx \Phi\boldsymbol{c}_i + \boldsymbol{w}_i\delta_i \tag{11}$$

for a vector $\boldsymbol{w}_i$ and scalar δ_i. Plug Eq. (11) into Eq. (4), set $\delta_i = 1$, and set $\lambda_i = 0$ to obtain the static mode approximation

$$K\boldsymbol{w}_i = \boldsymbol{f}_i - (K-\lambda_i M)\Phi\boldsymbol{c}_i \tag{12}$$

To partially compensate for errors introduced by Eq. (12), δ_i is chosen by the simple line search

$$\begin{aligned}\delta_i &= \arg\min(\delta_i \boldsymbol{w}_i)^T\left[\frac{1}{2}(K-\lambda_i M)\Phi\boldsymbol{c}_i + \delta_i\boldsymbol{w}_i) - \boldsymbol{f}_i\right]\\ \Rightarrow \delta_i &= \frac{\boldsymbol{w}_i^T\boldsymbol{f}_i}{\boldsymbol{w}_i^T(K-\lambda_i M)\boldsymbol{w}_i}\end{aligned} \tag{13}$$

D. Zhang and Zerva's Iterative Method

Although Wang's method[3] suffices in many cases, the lack of a fully accurate solution has motivated researchers to seek additional improvements. One recent development proposed by Zhang and Zerva[4] involves iterating on Wang's method. Zhang and Zerva eliminate the ad hoc parts of Wang's method, namely, the scaling factor δ_i and the selective omission of the term $\lambda_i M\boldsymbol{w}_i$. Although it is presented in a somewhat different manner in their paper,[4] Zhang and Zerva's technique can be interpreted as follows:

$$\boldsymbol{w}_i^{(0)} = 0 \tag{14a}$$

$$\bar{\boldsymbol{w}}_i^{(k)} = (I - \Phi\Phi^T M)\boldsymbol{w}_i^{(k)} \tag{14b}$$

$$K\boldsymbol{w}_i^{(k+1)} = \{\boldsymbol{f}_i - (K-\lambda_i M)\Phi\boldsymbol{c}_i\} + \lambda_i M\bar{\boldsymbol{w}}_i^{(k)} \tag{14c}$$

This result is very similar to Wang's method in that it employs a static solution. However, by adding the last term of the right-hand side of the last equation and iterating, at convergence the solution is

$$(K-\lambda_i M)(\Phi\boldsymbol{c}_i + \boldsymbol{w}_i) = \boldsymbol{f}_i \tag{15}$$

which exactly satisfies Eq. (4). The M orthogonalization in Eq. (14b) is included because the exact solution $\boldsymbol{w}_i$ is M orthogonal with respect to the columns of Φ.

IV. PCPG-Based Iterative Technique

The natural improvement of Zhang and Zerva's method is, instead of solving Eq. (14c) by iteration, to use the appropriate CG-type method to solve

$$(K - \lambda_i M)\boldsymbol{w}_i = \boldsymbol{f}_i - (K - \lambda_i M)\Phi \boldsymbol{c}_i \tag{16}$$

The problem with a CG approach to Eq. (16) is that the governing operator $(K - \lambda_i M)$ is indefinite, whereas the CG theory guarantees convergence only for a positive-definite operator. Thus, our first task is to define the governing operator and an associated preconditioner so that the advantages of the CG method can be retained in the present context.

A. Definition of the Matrix Operator and Preconditioner

If Ψ is defined as the matrix whose columns are the eigenvectors that were not computed, then let

$$\boldsymbol{w}_i = \Psi \boldsymbol{x}_i \tag{17}$$

Premultiply Eq. (16) by Ψ^T and substitute Eq. (17) to obtain the system

$$\left\{\Psi^T (K - \lambda_i M)\Psi\right\}\boldsymbol{x}_i = \Psi^T (\boldsymbol{f}_i - (K - \lambda_i M)\Phi \boldsymbol{c}_i) \tag{18}$$

Here, $\Psi^T (K - \lambda_i M)\Psi$ is positive definite as long as mode i and all modes below mode i have been computed. Therefore, it is possible to apply the PCG algorithm to Eq. (18). We choose the preconditioner

$$M_p = \Psi^T (K - \sigma M)\Psi \tag{19}$$

This choice is motivated by the potential availability of the factorization $LL^T = K - \sigma M$ from the solution of the eigenproblem $K\Phi = M\Phi\Omega^2$ via iterative methods such as the Lanczos algorithm.[9] Usually, $\sigma < \lambda_{\min}$ and is often chosen as $\sigma = 0$ for systems with no rigid-body modes or $\sigma < 0$ for systems with rigid-body modes. The effect of the preconditioner is to alter the condition number of the governing operator. The condition number of the deflated unconditioned operator is given by

$$\kappa_2 = \frac{(\lambda_{\max} - \lambda_i)}{(\lambda_{m+1} - \lambda_i)} \cong \frac{\lambda_{\max}}{(\lambda_{m+1} - \lambda_i)} \tag{20}$$

whereas the condition number of the deflated preconditioned operator is

$$\kappa_2 = \left(\frac{\lambda_{\max} - \lambda_i}{\lambda_{\max} - \sigma}\right) \Big/ \left(\frac{\lambda_{m+1} - \lambda_i}{\lambda_{m+1} - \sigma}\right) \cong \frac{1}{[1 - (\lambda_i / \lambda_{m+1})]} \tag{21}$$

Thus, as $\lambda_{\max}$ becomes large, which is true for all reasonably accurate and detailed computational models, the original problem, even with deflation, becomes highly ill conditioned. The preconditioned problem, on the other hand, has very good numerical properties except in the limit as the eigenvalue of the mode of interest approaches the lowest eigenvalue of the residual spectrum.

B. Derivation of the PCPG-Based Iterative Technique

The implementation of CG preconditioned as in Eq. (19) is as follows.[9]

1. Initialize $k = 0 \quad \boldsymbol{x}_0 = 0 \quad r_0 = \Psi^T\{\boldsymbol{f}_i - (K - \lambda_i M)\Phi \boldsymbol{c}_i\}$
2. While $(\boldsymbol{r}_k \neq 0)$:
   2a. Solve $\Psi^T (K - \sigma M)\Psi \boldsymbol{z}_k = \boldsymbol{r}_k$
   2b. Increment $k = k + 1$
   2c. If $k = 1$, $p_1 = z_0$, else

$$\begin{bmatrix} \beta_k = \dfrac{\boldsymbol{r}_{k-1}^T \boldsymbol{z}_{k-1}}{\boldsymbol{r}_{k-2}^T \boldsymbol{z}_{k-2}} \\ \boldsymbol{p}_k = \boldsymbol{z}_{k-1} + \beta_k \boldsymbol{p}_{k-1} \end{bmatrix} \tag{22}$$

   2d. $\alpha_k = \dfrac{\boldsymbol{r}_{k-1}^T \boldsymbol{z}_{k-1}}{\boldsymbol{p}_k^T \left\{\Psi^T (K - \lambda_i M)\Psi\right\}\boldsymbol{p}_k}$
   2e. $\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \alpha_k \boldsymbol{p}_k$
   2f. $\boldsymbol{r}_k = \boldsymbol{r}_{k-1} - \alpha_k \{\Psi^T (K - \lambda_i M)\Psi\}\boldsymbol{p}_k$
   2g. end while

This implementation is not practical, however, because Ψ is not practical to compute. As we shall see, it is possible to apply PCG without explicitly computing Ψ. Inverting the identity, $[\Phi, \Psi]^T M[\Phi, \Psi] = I$, we obtain $M^{-1} = [\Phi, \Psi][\Phi, \Psi]^T = \Phi\Phi^T + \Psi\Psi^T$. The trick is to use this expansion to compute the action of $\Psi\Psi^T$. First, we make the following definitions:

$$\bar{\boldsymbol{r}}_k = M\Psi \boldsymbol{r}_k \qquad \boldsymbol{q}_k = \Psi \boldsymbol{p}_k \qquad \boldsymbol{y}_k = \Psi \boldsymbol{z}_k \tag{23}$$

We can solve the preconditioner equation (noting that $\boldsymbol{r}_k = \Psi^T \bar{\boldsymbol{r}}_k$) as

$$\begin{aligned} &\Psi^T (K - \sigma M)\Psi \boldsymbol{z}_k = \Psi^T \bar{\boldsymbol{r}}_k \\ &\therefore (K - \sigma M)\Psi \boldsymbol{z}_k = M\Psi\Psi^T \bar{\boldsymbol{r}}_k \end{aligned} \tag{24}$$

Thus,

$$\begin{aligned} \Psi \boldsymbol{z}_k &= (K - \sigma M)^{-1} M\Psi\Psi^T \bar{r}_k \\ &= (K - \sigma M)^{-1} (I - M\Phi\Phi^T)\bar{\boldsymbol{r}}_k \end{aligned} \tag{25}$$

by using the identity $M^{-1} = \Phi\Phi^T + \Psi\Psi^T$. Then, we can write Eq. (25) in terms of $\boldsymbol{y}_k$, viz.,

$$\boldsymbol{y}_k = (I - \Phi\Phi^T M)\Psi z_k = (I - \Phi\Phi^T M)(K - \sigma M)^{-1} (I - M\Phi\Phi^T)\bar{\boldsymbol{r}}_k \tag{26}$$

where the filtering operator $(I - \Phi\Phi^T M)$ ensures that $\boldsymbol{y}_k$ remains in the subspace of Ψ as defined. The remainder of the algorithm involves the variable substitutions defined in Eqs. (17) and (23) to obtain

1. Initialize $k = 0 \qquad \boldsymbol{w}_0 = 0 \qquad \bar{\boldsymbol{r}}_0 = \{\boldsymbol{f}_i - (K - \lambda_i M)\Phi \boldsymbol{c}_i\}$
2. While $(\bar{\boldsymbol{r}}_k \neq 0)$:
   2a. Solve $(K - \sigma M)\bar{y}_k = (I - M\Phi\Phi^T)\bar{\boldsymbol{r}}_k$
   2b. Project $\boldsymbol{y}_k = (I - \Phi\Phi^T M)\bar{y}_k$
   2c. Increment $k = k + 1$
   2d. If $k = 1$, $q_1 = y_0$, else

$$\begin{bmatrix} \beta_k = \dfrac{\bar{\boldsymbol{r}}_{k-1}^T \boldsymbol{y}_{k-1}}{\bar{\boldsymbol{r}}_{k-2}^T \boldsymbol{y}_{k-2}} \\ \boldsymbol{q}_k = y_{k-1} + \beta_k \boldsymbol{q}_{k-1} \end{bmatrix} \tag{27}$$

   2e. $\alpha_k = \dfrac{\bar{\boldsymbol{r}}_{k-1}^T \boldsymbol{y}_{k-1}}{\boldsymbol{q}_k^T (K - \lambda_i M)\boldsymbol{q}_k}$
   2f. $\boldsymbol{w}_k = \boldsymbol{w}_{k-1} + \alpha_k \boldsymbol{q}_k$
   2g. $\bar{\boldsymbol{r}}_k = \bar{\boldsymbol{r}}_{k-1} - \alpha_k (K - \lambda_i M)\boldsymbol{q}_k$
   2h. end while

The projections utilized (which could simply be considered integral to the preconditioner) are the reason this method is termed a PCPG algorithm, after the example of Farhat and Roux.[10]

C. Convergence Properties of the PCPG-Based Iterative Technique

The convergence properties of the present algorithm are derived from the well-known convergence theorems for the CG algorithm.[9] Specifically, given the preconditioner defined earlier and the projections that deflate the governing operator, the condition number (assuming $M = I$ for simplicity) is

$$\kappa_2 = \left(\frac{\lambda_{\max} - \lambda_i}{\lambda_{\max} - \sigma}\right) \Big/ \left(\frac{\lambda_{m+1} - \lambda_i}{\lambda_{m+1} - \sigma}\right) \tag{28}$$

and the solution error at iteration k decreases as

$$\left\|\boldsymbol{e}_k^{[\mathrm{PCPG}]}\right\|_A^2 \leq [(\sqrt{\kappa_2} - 1)/(\sqrt{\kappa_2} + 1)]^{2k} \|\boldsymbol{e}_0\|_A^2 \tag{29}$$

where the solution error at iteration k is defined as $\boldsymbol{e}_k^{[\mathrm{PCPG}]} = (\boldsymbol{w} - \boldsymbol{w}_k^{[\mathrm{PCPG}]})$ and the norm $\|(\,)\|_A$ is defined as

$$\|\boldsymbol{v}\|_A = \sqrt{\boldsymbol{v}^T A \boldsymbol{v}} \qquad A = (K - \lambda_i M) \tag{30}$$

D. Summary of the Present PCPG-Based Iterative Technique

In summary, the iterative procedure for computing eigenvector derivatives is as follows.

1) Solve the nominal eigenproblem $K\Phi = M\Phi\Lambda$ for the lowest m eigenvalues/eigenvectors. As part of this analysis, we must typically form the operator $(K - \sigma M)$. We choose $\sigma < \lambda_{\min}$ such that the operator is positive definite and admits the Cholesky decomposition $(K - \sigma M) = LL^T$.

2) Compute the eigenvalue derivatives via Eq. (3) and the modal superposition contribution $\Phi \boldsymbol{c}_i$ to the eigenvector derivative, i.e., Fox's method, by Eq. (10).

3) Apply PCPG algorithm as given by Eq. (27) to obtain $\boldsymbol{w}_i$.

4) Add to modal superposition solution to obtain final result: $\phi_i' = \Phi \boldsymbol{c}_i + \boldsymbol{w}_i$.

V. Evaluation of PCPG-Based Iterative Technique

In the examples to follow, two finite element models are considered. The first example is a 696-degree-of-freedom (DOF) welded tubular aluminum frame (model A). The structure was modeled in MSC/NASTRAN with bar elements, and the lowest 25 modes were computed, including 6 rigid-body modes and 19 flexible modes. The second example, model B, is a 3649-DOF model of an automotive body-in-white. This structure was also modeled in MSC/NASTRAN with bar and plate elements, and the lowest 25 modes were computed, again including 6 rigid-body modes and 19 flexible modes. The distributions of the computed frequencies for the two models are given in Table 1, plus the frequency of the first mode of the residual mode set of each model.

For evaluation purposes, we first compare the present iterative technique to Fox's method,[1] Wang's method,[3] and Nelson's method.[2] This is accomplished by applying the same algorithm for all of the approximate methods and limiting the maximum number of iterations to 0 for Fox's method, and to 1 for Wang's method. The new method results were limited to 10 iterations. The approximate solutions were then compared to the direct exact solution obtained using Nelson's method.

A. Comparison of Accuracy Between Incomplete Approximations and New Iterative Method

The model used in the first comparison is model A, for which nine design parameters were considered, including six joint compliance elements and three global beam cross-sectional properties. The goal of the analysis is to determine the eigenvector derivatives for all 19 flexible modes using modal superposition, Fox's method, and the new PCPG-based iterative approach. Figure 1 illustrates the absolute error in Eq. (4) for all four methods. The values plotted are the mean of the error norms for each of the flexible modes across the nine design parameters. It is clear that the present algorithm is effective in reducing the error in the incomplete modal superposition solution and improves significantly on Wang's method of using a single static correction. Additional iterations on the new technique would reduce these errors further, as will be seen in subsequent examples.

Table 1 Computed frequencies of models used in example problems

	Frequency, Hz	
Mode no.	Model A, $n = 696$	Model B, $n = 3649$
R1–R6	~0.0	~0.0
1	88.5	22.45
2	175.0	25.97
3	219.8	36.57
4	228.2	38.65
5	268.2	40.73
6	324.0	44.09
7	334.7	48.90
8	367.8	50.67
9	382.6	55.59
10	455.5	59.52
11	540.1	61.26
12	633.5	66.74
13	731.8	66.90
14	900.6	69.15
15	906.3	73.10
16	959.1	77.70
17	1052.6	89.26
18	1069.0	89.78
19	1189.7	92.61
20	1201.7	92.77

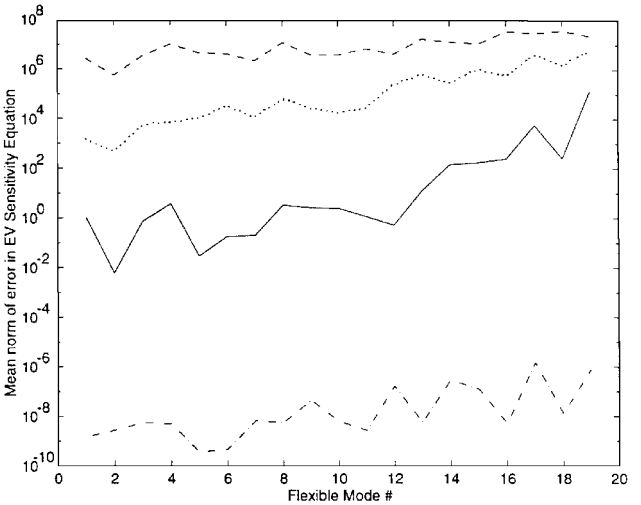

Fig. 1 Comparison of errors in eigenvector derivative equation: – – –, modal superposition; ····, Wang's method[3]; ——, new iterative technique; and –·–, Nelson's exact method.[2]

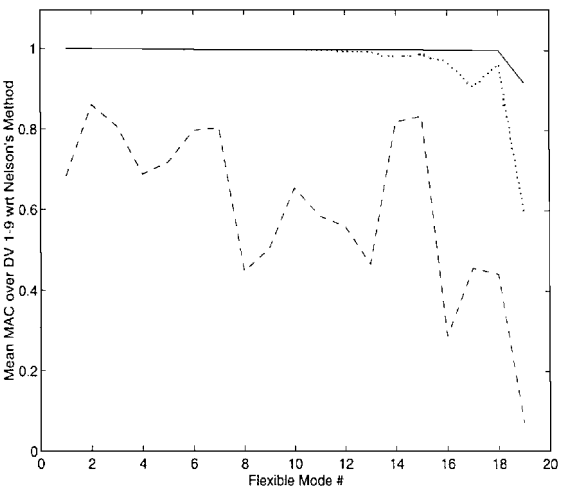

Fig. 2 Comparison of shape errors for eigenvector derivatives: – – –, modal superposition; ····, Wang's method[3]; and ——, new iterative technique.

Having established the relative accuracy of the new iterative method as compared to Fox's method and Wang's static correction, we now investigate the relative difference in the actual sensitivity vectors computed. There are many ways to compare the vectors results, and in the present application there are many results to consider. Specifically, there is an eigenvector derivative, which is a vector quantity, for each design variable and each mode in the analysis. To simplify the comparisons, we can examine the relative correlation of the vectors using the modal assurance criteria (MAC),[11] defined for two vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ as

$$\mathrm{MAC}(\boldsymbol{v}_1, \boldsymbol{v}_2) = \frac{\left(\boldsymbol{v}_1^T \boldsymbol{v}_2\right)^2}{\left(\boldsymbol{v}_1^T \boldsymbol{v}_1\right)\left(\boldsymbol{v}_2^T \boldsymbol{v}_2\right)} \tag{31}$$

When the vectors are nearly parallel, the MAC measure will be nearly 1.0. Note that this measure normalizes out any differences in scale, which are not arbitrary for these derivative quantities. Thus, a MAC that is close to 1.0 is a necessary but not a sufficient condition for agreement between the vector results.

Figure 2 shows the MAC results between the exact solution and the approximate solutions via modal superposition, Wang's method,

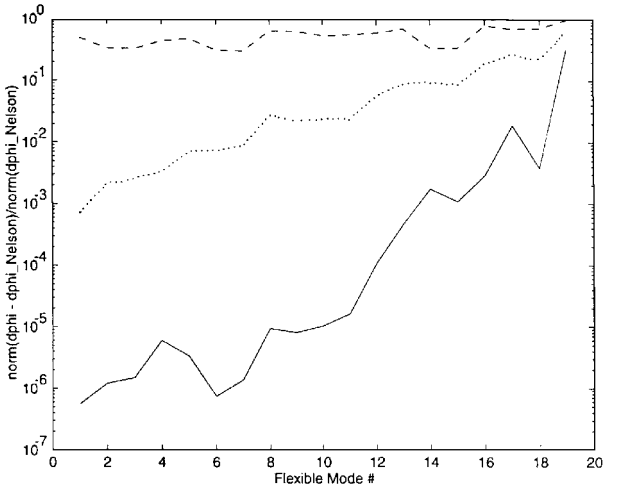

Fig. 3 Comparison of errors in eigenvector derivative solutions: – – –, modal superposition; ····, Wang's method[3]; and ——, new iterative technique.

and the new PCPG-based iterative technique for flexible modes 1–19. The mean of the MAC over the nine design parameters is plotted for comparison. Note that the modal superposition solution exhibits large errors even for the lowest modes. Thus, even if the analyst uses a large set of nominal modes to compute the modal superposition solution, there is no guarantee that the eigenvector derivatives for the lowest modes will be sufficiently accurate. From the results for Wang's method, it would appear that a single static correction provides perhaps sufficient accuracy for the lowest modes of the system. However, the higher modes near the limit of the computed spectrum are still in error due to the incompleteness of the static correction. Finally, the results for the new PCPG-based iterative technique show that, after 10 iterations, the sensitivities for all but the highest modes have nearly converged to the exact solution.

Finally, Fig. 3 compares the error between the exact and approximate solutions (including both relative shape and scaling) normalized to the magnitude of the exact solution. Again, the errors for the modal superposition solution are high, ranging from 30% to nearly 100% of the correct solution. The relative errors for Wang's method range from below 1% for modes 1–7 to as high as 25% for modes 17 and 18 and 50% for mode 19. The relative error for the new iterative technique is below 1% for all modes except mode 17 (about 2%) and 19 (about 30%).

B. Comparison of Computational Costs Between PCPG and Nelson's Method

To demonstrate the efficiency of the iterative PCPG method, it is important to quantify the computational cost for both the iterative approach and for the direct exact method of Nelson. For comparison to the direct solution, we have computed the derivatives with respect to the lowest 25 eigenvectors for a single design parameter. The computation and memory requirements given subsequently are dependent on the specific problems considered and the implementation of the algorithms. For both methods, the numerical methods were implemented using sparse Matlab[12] matrix methods for factorizations, backward solutions, dot products, and so forth. The model size and matrix sparsity information is given in Tables 2 and 3. The results for model A are given in Table 2. The stopping criterion used for the PCPG iterations is

$$\|\bar{\boldsymbol{r}}_k\| < 1 \times 10^{-6}\|\boldsymbol{f}_i\| \tag{32}$$

The design parameter chosen is the localized rotational compliance of a welded joint. Note that the direct exact solution for this model is about twice as efficient as the iterative technique. This is due to the relatively small size of the model. As the order of the problem increases, however, the cost of factorizing the indefinite operator $K - \lambda_i M$ for each mode in Nelson's method quickly surpasses that of all of the operations in the PCPG method. This is seen clearly in the results for model B, which are given in Table 3. Here

Table 2 Computational comparison, PCPG vs Nelson's method[2]: model A (n = 696, K sparsity 0.98% full, memory LU/ Cholesky matrix decomposition = 1.07)

Performance metric	Nelson, MFLOPs	PCPG	
		No. iterations	MFLOPs
Average per mode	0.425	4	0.953
Best case per mode	——	1	0.374
Worst case per mode	——	11	2.472
Total, 25 modes	10.63	——	23.83[a]

[a]24.02 MFLOPs including factorization of $(K - \sigma M)$.

Table 3 Computational comparison, PCPG vs Nelson's method[2]: model B (n = 3649, K sparsity 1.12%, memory LU/ Cholesky matrix decomposition = 7.38)

Performance metric	Nelson, MFLOPs	PCPG	
		No. iterations	MFLOPs
Average per mode	2,832	7	16.4
Best case per mode	——	2	5.28
Worst case per mode	——	24	50.9
Total, 25 modes	70,800[a]	——	409.9[b]

[a]Estimated. [b]433.4 MFLOPs including factorization of $(K - \sigma M)$.

the floating-point operations (FLOPs) required to compute a single eigenvector sensitivity via Nelson's method is about 7 times that required to compute all 25 eigenvectors using the PCPG method. Note also that the memory required to store the upper-triangular factor of $(K - \lambda_i M)$ for Nelson's method is over seven times the memory requirement for the Cholesky factorization of the preconditioner $(K - \sigma M)$. Finally, note that the cost of computing the Cholesky factorization is not included in the average and total FLOPs counts in Tables 2 and 3 because it is already available if the algorithm is applied immediately after the computing the nominal eigensolution. The total cost with the Cholesky factorization is given at the bottom of Tables 2 and 3. For model B, retaining the factorization of the preconditioner in memory from the nominal eigensolution saves 23.5×10^6 FLOPs (MFLOPs), which is 5.4% of the total computations for the sensitivity analysis. Relatively larger savings can be expected for models with greater numbers of DOF.

C. Convergence Rates for Iterative Methods

As a final demonstration of the efficiency of the present technique, we compare the present PCPG method to Zhang and Zerva's iterative method.[4] In both methods, the underlying minimization problem involves the indefinite operator $(K - \lambda_i M)$, a property that is known to be problematic for iterative-based solvers. The PCPG-based technique uses the filter operator $(I - \Phi\Phi^T M)$ to project out unwanted components in the solution and the residual. In Zhang and Zerva's technique, the same filtering operator is applied, but only to the solution vector $\boldsymbol{w}_k$. Furthermore, their method does not employ the search direction orthogonality, which is key to CG methods. Therefore, we expect some difference in convergence rates, particularly when computing sensitivities at the limit of the computed modal spectrum. For Zhang and Zerva's method, it was also necessary to use the shifted operator because of the singularity of the stiffness matrix. This is easily accomplished by replacing Eq. (14c) by

$$(K - \sigma M)\boldsymbol{w}_i^{(k+1)} = \{\boldsymbol{f}_i - (K - \lambda_i M)\Phi\boldsymbol{c}_i\} + (\lambda_i - \sigma)M\boldsymbol{w}_i^{-(k)} \tag{33}$$

The convergence for Zhang and Zerva's method for $\sigma = 0$ is given by

$$\boldsymbol{e}_k^{[ZZ]} = \sum_{j=m+1}^{n} \left(\frac{\lambda_i}{\lambda_j}\right)^k \left(\frac{\phi_j^T \boldsymbol{f}_i}{\lambda_j - \lambda_i}\right)\phi_j \tag{34}$$

which leads to the bound

$$\left\|\boldsymbol{e}_k^{[ZZ]}\right\|_A^2 \le \left(\frac{\lambda_i}{\lambda_{m+1}}\right)^{2k} \|\boldsymbol{e}_0\|_A^2 \tag{35}$$

corresponding to Eq. (29). Using the convergence property for the PCPG-based algorithm [Eq. (29)], setting $\sigma = 0$ for comparison

Table 4 Comparison of iterative algorithms based on convergence formulas

$\frac{\lambda_i}{\lambda_{m+1}}$	$\left[\frac{\lambda_i}{\lambda_{m+1}}\Big/\left(1+\sqrt{1-\frac{\lambda_i}{\lambda_{m+1}}}\right)\right]$	Solution error measure $\|e_k\|_A^2$ after 10 iterations (relative to $\|e_0\|_A^2$)		Number of iterations required for $\|e_k\|_A^2 \le 1 \times 10^{-10}\|e_0\|_A^2$	
		Zhang–Zerva[4]	PCPG	Zhang–Zerva	PCPG
0.1000	0.0513	1.00e−20	1.60e−26	5	4
0.5000	0.2929	9.5e−07	2.16e−11	17	10
0.8000	0.5527	0.01153	7.08e−06	52	20
0.9000	0.6838	0.12158	5.00e−04	110	31
0.9900	0.9000	0.81791	0.12158	1,146	110
0.9990	0.9684	0.98019	0.52613	11,508	359

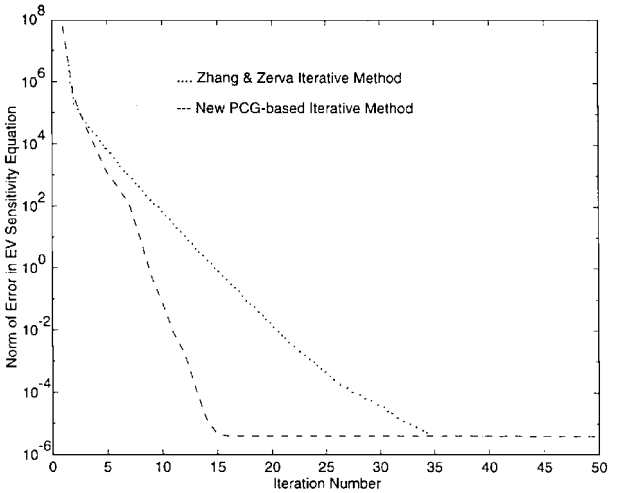

Fig. 4 Convergence rates: design parameter 7, flexible mode 14.

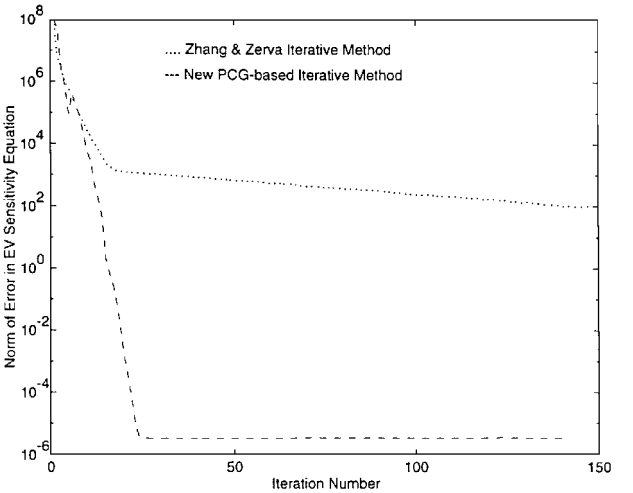

Fig. 5 Convergence rates: design parameter 7, flexible mode 19.

to Zhang and Zerva's technique, and assuming $\lambda_{\max} \gg \lambda_{m+1}$, we have

$$\kappa \approx \frac{1}{[1-(\lambda_i/\lambda_{m+1})]} \tag{36}$$

$$\left\|e_k^{[\mathrm{PCPG}]}\right\|_A^2 \le \left[\frac{\lambda_i}{\lambda_{m+1}}\Big/\left(1+\sqrt{1-\frac{\lambda_i}{\lambda_{m+1}}}\right)\right]^{2k} \|e_0\|_A^2 \tag{37}$$

Comparing Eqs. (35) and (37), the present PCPG-based algorithm displays superior convergence properties for all $\{\lambda_i, i = 1, \ldots, m\}$. For example, Table 4 compares the two iterative algorithms based on the convergence analysis just given. Note that, as the eigenvalue λ_i for the mode under consideration approaches the eigenvalue of the first mode in the residual set λ_{m+1}, the convergence rates of both algorithms degrade significantly.

Some characteristic numerical results from model A are shown in Figs. 4 and 5, which support the conclusions based on the theoretical convergence analysis. For this example problem, the shift point is $\sigma = -1000$. For flexible mode 14 ($\lambda_i/\lambda_{m+1} = 0.562$), both methods converge to the same accuracy limit, although the convergence rate of the PCPG-based technique is about twice that of Zhang and Zerva's method. For flexible mode 19 ($\lambda_i/\lambda_{m+1} = 0.980$), which is at the limit of the spectrum computed by the nominal eigensolver, the convergence differences are even more dramatic. The PCPG-based technique still converges to the accuracy limit in approximately 20 iterations, whereas the convergence rate of Zhang and Zerva's method degrades at about 15 iterations and remains with a relative error of 10^7 with respect to the accuracy limit after 150 iterations.

The dependence of the convergence on the mode number and model order for a single design variable is demonstrated in Fig. 6. Again, note that the convergence rate for both models degrades as the mode number increases. This is due to the relative difference between the eigenvalue of the mode for which sensitivities are being computed and the eigenvalues of the modes contributing

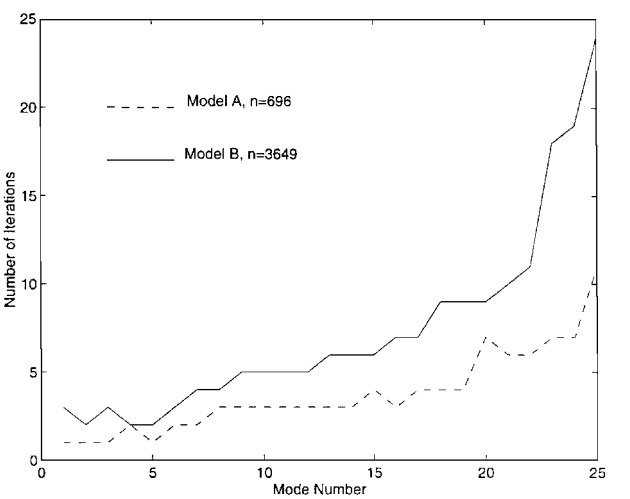

Fig. 6 Convergence rate for PCPG over modal spectrum for models A and B.

to the residual quantity w_i. As this difference becomes small, i.e., $\lambda_i \to \lambda_{m+1}$, which occurs when computing the sensitivities for the highest calculated eigenmodes, the convergence rate can slow down considerably as predicted by the convergence analysis. A reasonable remedy for this condition is to extend the computed eigenspectrum in the nominal eigenvalue analysis beyond the highest mode for which sensitivities are to be computed. Whereas this strategy will reduce the number of iterations in the PCPG method when computing the sensitivities of the highest modes, it may not reduce the total number of computations or the CPU time spent in the analysis because of the cost of computing and utilizing the additional modes. It is generally prudent to compute at least one additional mode beyond the highest mode for which sensitivities will be computed, to ensure that there is a reasonable separation in frequency between the modes of sensitivity interest and the modes of the residual spectrum. That is, we should know the value of λ_{i+1} for all i so that the

critical convergence ratio λ_i/λ_{m+1} is at least bounded by the known quantity λ_i/λ_{i+1}.

VI. Conclusions

An iterative procedure has been presented for computing modal vector sensitivities due to finite element model parameter variations. The present method is a PCPG-based technique and is intended to utilize the existing matrix factorizations developed for an iterative eigensolution such as Lanczos or Subspace Iteration. As such, this technique can be integrated into a coupled eigensolver/sensitivity software module and can leverage off the nonrecurring costs of the solver. The two model examples provided demonstrate both the accuracy of the present technique and its superior efficiency as compared to existing techniques with similar accuracy. The present technique, therefore, is recommended for eigenvector derivative computations for finite element models with more than 1000 DOF. Extensions of the present technique to the problem of repeated modes are currently being considered and will hopefully be demonstrated in the near future.

Acknowledgment

This work was supported by the U.S. Department of Energy under Contract DE-AC04-94AL85000. Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy. The author would like to thank David Day, John Red-Horse, and Michael Eldred of Sandia National Laboratories and K. C. Park and Charbel Farhat of the University of Colorado for their interest and encouragement.

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. 14, No. 9, 1976, pp. 1201–1205.

[3]Wang, B. P., "Improved Approximate Methods for Computing Eigenvector Derivatives in Structural Dynamics," *AIAA Journal*, Vol. 29, No. 6, 1991, pp. 1018–1020.

[4]Zhang, O., and Zerva, A., "Iterative Method for Calculating Derivatives of Eigenvectors," *AIAA Journal*, Vol. 34, No. 5, 1996, pp. 1088–1090.

[5]Beliveau, J.-G., Cogan, S., Lallement, G., and Ayer, F., "Iterative Least-Squares Calculation for Modal Eigenvector Sensitivity," *AIAA Journal*, Vol. 34, No. 2, 1996, pp. 385–391.

[6]Ojalvo, I. U., "Efficient Computation of Modal Sensitivities for Systems with Repeated Frequencies," *AIAA Journal*, Vol. 26, No. 3, 1988, pp. 361–366.

[7]Bernard, M. L., and Bronowicki, A. J., "Modal Expansion Method for Eigensensitivity with Repeated Roots," *AIAA Journal*, Vol. 32, No. 7, 1994, pp. 1500–1506.

[8]Mills-Curran, W. C., "Calculation of Eigenvector Derivatives for Structures with Repeated Eigenvalues," *AIAA Journal*, Vol. 26, No. 7, 1988, pp. 867–871.

[9]Golub, G. H., and Van Loan, C. F., *Matrix Computations*, 2nd ed., Johns Hopkins Univ. Press, Baltimore, MD, 1989.

[10]Farhat, C., and Roux, F. X., "A Method of Finite Element Tearing and Interconnecting and its Parallel Solution," *International Journal for Numerical Methods in Engineering*, Vol. 32, 1991, pp. 1205–1227.

[11]Ewins, D. J., *Modal Testing: Theory and Practice*, Research Studies Press, Ltd., Somerset, England, UK, 1984, pp. 225, 226.

[12]"Matlab User's Guide," MathWorks, Natick, MA, 1993.

A. Berman
Associate Editor