An Iterative Method for Eigenvector Derivatives

Gerald D. High
The MacNeal-Schwendler Corporation
Los Angeles, CA

To be presented at the MSC World Users Conference
"Integration in the CAE Environment"
March 26-30, 1990

Abstract

An iterative method has been developed for calculating eigenvector derivatives. The method, called the Iterative Modal Method (IMM), combines the modal method of Fox and the iterative method of Rudisill and Chu. Only one decomposition is required, regardless of how many design variables or eigenvectors there are in the problem. IMM has been implemented in Version 66 of MSC/NASTRAN. A comparision is made with Nelson's method in two example problems. IMM is shown to be cheaper than Nelson's method on the larger of the two example problems.

Nomenclature

Δb = perturbation in design variable b
$\lceil I \rfloor$ = identity matrix
$[K_o]$ = baseline stiffness matrix
$[\Delta K^b]$ = perturbation in $[K_o]$ due to change in design variable b
[K] = perturbed stiffness matrix
$\lceil \lambda_o \rfloor$ = baseline eigenvalue matrix (diagonal)
$\lceil \Delta\lambda^b \rfloor$ = perturbation in $[\lambda_o]$ due to change in design variable b
$\lceil \lambda \rfloor$ = perturbed eigenvalue matrix (diagonal)
$[M_o]$ = baseline mass matrix
$[\Delta M^b]$ = perturbation in $[M_o]$ due to change in design variable b
[M] = perturbed mass matrix
$[\phi_o]$ = baseline eigenvector matrix (mode shapes)
$[\Delta\phi^b]$ = perturbation in $[\phi_o]$ due to change in design variable b
$[\phi]$ = perturbed eigenvector matrix (mode shapes)
[R] = Gram-Schmidt orthogonalization matrix
$[X^b]$ = eigenvector change coefficient matrix

Introduction

Since the late 1960's several methods have been developed for calculating eigenvector derivatives[1-4]. Fox's modal method,[1] while relatively cheap, suffers from inaccuracy when a reduced set of modes is used. An exact method developed by Nelson[3] requires a matrix decomposition for each mode of interest and hence can be relatively expensive. By combining both Fox's method and Rudisill's and Chu's[2] iterative method, multiple matrix decompositions can be avoided. By using vectors calculated by Fox's method as a starting point, along with Gram-Schmidt orthogonalization, convergence is achieved for all the modes of interest (not just the lowest mode). The iterative method developed here requires only one decomposition, regardless of how many design variables or eigenvectors there are. For larger problems a cost savings is expected over Nelson's method.

Proofs on the convergence of Rudisill's iterative method are given by Andrew[5]. Andrew[5] and Tan[6] have also developed some additional refinements to speed up Rudisill's method.

Matrix Derivation of Fox's Method

Following the matrix derivation by Herting[4] the general eigenvalue equation is:

$$[M][\phi]\lceil\lambda\rfloor - [K][\phi] = 0 \qquad (1)$$

Let the perturbed quantities be given by

$$\begin{aligned} [M] &= [M_o] + [\Delta M^b] \qquad (2) \\ [K] &= [K_o] + [\Delta K^b] \\ \lceil\lambda\rfloor &= \lceil\lambda_o\rfloor + \lceil\Delta\lambda^b\rfloor \\ [\phi] &= [\phi_o] + [\phi_o][X^b] \end{aligned}$$

Note that it is assumed that the new eigenvectors can be represented as a linear combination of the old eigenvectors with the coefficient matrix $[X^b]$. Substituting (2) into

(1), subtracting the original solution (1), and dropping higher order terms yields:

$$\begin{aligned}[M_o][\phi_o]\lceil\Delta\lambda^b\rfloor + [M_o][\phi_o][X^b]\lceil\lambda_o\rfloor \quad (3)\\ +[\Delta M^b][\phi_o]\lceil\lambda_o\rfloor - [\Delta K^b][\phi_o]\\ -[K_o][\phi_o][X^b] = 0\end{aligned}$$

Define

$$\begin{aligned}\lceil I\rfloor &= [\phi_o]^T[M_o][\phi_o] \quad (4)\\ \lceil\lambda_o\rfloor &= [\phi_o]^T[K_o][\phi_o]\\ [\Delta m^b] &= [\phi_o]^T[\Delta M^b][\phi_o]\\ [\Delta k^b] &= [\phi_o]^T[\Delta K^b][\phi_o]\\ [D^b] &= [\Delta k^b] - [\Delta m^b]\lceil\lambda_o\rfloor\end{aligned}$$

Premultiply (3) by $[\phi]^T$ and substitute using (4) to obtain the first order equation:

$$[\Delta\lambda^b] + [X^b][\lambda_o] - [\lambda_o][X^b] = [D^b] \quad (5)$$

Solution for eigenvalue derivatives

The diagonal terms in (5) yields the eigenvalue perturbations (since the 2nd and 3rd terms cancel each other out on the diagonal)

$$\Delta\lambda_i^b = D_{ii}^b \quad (6)$$

or the derivatives

$$\frac{\partial\lambda_i^b}{\partial b} \cong \frac{\Delta\lambda_i^b}{\Delta b} = \frac{D_{ii}^b}{\Delta b} \quad (7)$$

Solution for eigenvector derivatives

To form the eigenvector derivatives the coefficient matrix $[X^b]$ must first be calculated. Eq. (5) may be solved for the off-diagonal elements of $[X^b]$ by noting that $\Delta\lambda_{ij}^b = 0 (i \neq j)$. Hence, for $(i \neq j)$:

$$\begin{aligned}X_{ij}^b &= \frac{D_{ij}^b}{\lambda_j - \lambda_i} \quad (8)\\ &= \frac{\Delta k_{ij}^b - \Delta m_{ij}^b\lambda_{oj}}{\lambda_j - \lambda_i}\end{aligned}$$

Note that (8) is not valid for identical roots and $[X^b]$ is undefined. Note also that in general no conclusions can be made whether $[X^b]$ is symmetric, unsymmetric or skew symmetric (although $[\Delta k^b]$ and $[\Delta m^b]$ are symmetric the same can not be said for the product $[\Delta m^b]\lceil\lambda_o\rfloor$).

Solution for the diagonal values of $[X^b]$ can be obtained by the constraint that the perturbed modes must also be M-orthonormal.

$$\begin{aligned}\lceil I\rfloor = ([\phi_o] + [\phi_o][X^b])^T[M_o + \Delta M^b] \quad (9)\\ ([\phi_o] + [\phi_o][X^b]) =\\ [I + X^b]^T[\phi_o]^T[M + \Delta M^b][\phi_o][I + X^b]\end{aligned}$$

Simplifying yields:

$$[0] = [X^b] + [X^b]^T + [\Delta m^b] \quad (10)$$

From (10) the diagonal values are

$$X_{ii}^b = -\frac{\Delta m_{ii}^b}{2} \quad (11)$$

In summary, the eigenvector change matrix $[X^b]$ is defined by:

$$X_{ij} = \begin{cases} -\frac{\Delta m_{ii}^b}{2} & \text{if i = j}\\ \frac{D_{ij}^b}{\lambda_j - \lambda_i} & \text{if i} \neq \text{j}\end{cases} \quad (12)$$

The resulting change in ϕ_o is

$$[\Delta\phi^b] = [\phi_o][X^b] \tag{13}$$

and the derivative by

$$\frac{[\partial\phi]}{\partial b} \cong \frac{[\Delta\phi^b]}{\Delta b} = \frac{[\phi_o][X^b]}{\Delta b} \tag{14}$$

Iterative Modal Method

Rewriting Eq. (3) in terms of $[\Delta\phi^b]$ and rearranging yields:

$$[K_o][\Delta\phi^b] = [M_o][\Delta\phi^b]\lceil\lambda_o\rfloor + [F] \tag{15}$$

where

$$[F] = [M_o][\phi_o][\Delta\lambda^b] + [\Delta M^b][\phi_o][\lambda_o] - [\Delta K^b][\phi_o] \tag{16}$$

The only unknown in (15) is $[\Delta\phi^b]$, which appears in both the left and right sides. Eq. (15) will not be satisfied by (13) if a reduced set of eigenvectors is used. However, using $[\Delta\phi^b] = [\phi][X^b]$ as the starting solution, (15) may be solved iteratively. Eq. (16)may also be considered a static pseudo load, analogous to the mode-acceleration [7,8] method in structural dynamics.

Rudisill developed a similiar iterative equation based on a similiar system as (1). However his method required the derivatives of the matrix product $[K_o]^{-1}[M_o]$.

Gram-Schmidt Orthogonalization

A Gram-Schmidt orthogonalization along with renormalization must be performed at each iteration on $[\Delta\phi^b]$. The Gram-Schmidt orthogonalization for $[\phi]$ is

$$[\phi] = [\phi_o][R] \tag{17}$$

where $[R]$ is defined to be:

$$R_{ij} = \begin{cases} 1 & \text{if i = j} \\ 0 & \text{if i > j} \\ -\{\phi_i\}^T[M]\{\phi_j\} & \text{if i < j} \end{cases} \tag{18}$$

Perturbation of (17) yields:

$$[\Delta\phi^b] = [\Delta\phi_o^b][R] + [\phi_o][\Delta R^b] \tag{19}$$

where $[\Delta R^b]$ is:

$$\Delta R_{ij}^b = \begin{cases} 0 & \text{if i} \geq \text{j} \\ -\{\Delta\phi_i^b\}^T[M]\{\phi_j\}- & \\ \{\phi_i\}^T[\Delta M^b]\{\phi_j\}- & \\ \{\phi_i\}^T[M]\{\Delta\phi_j^b\} & \text{if i < j} \end{cases} \tag{20}$$

Mass Normalization

To prevent $[\Delta\phi^b]$ from changing magnitude during each iteration, mass normalization is performed. For the ith perturbed mode, the scale factor α_i can be calculated from:

$$1 = \left(\{\phi_o\}_i + \alpha_i\{\Delta\phi^b\}_i\right)^T [M_o + \Delta M^b] \left(\{\phi_o\}_i + \alpha_i\{\Delta\phi^b\}_i\right) \tag{21}$$

Simplifying gives the scale factor for $\{\Delta\phi^b\}_i$ as

$$\alpha_i = -\frac{\{\phi_o\}_i^T[\Delta M^b]\{\phi_o\}_i}{2\{\phi_o\}_i^T[M_o]\{\Delta\phi^b\}_i} \tag{22}$$

Convergence Criterion

The error of the algorithm at some iteration i may be measured by the square of the Euclidean norm of (15) over each column:

$$\epsilon_i = max(\|[K_o][\Delta\phi^b]_i - [F] - [M_o][\Delta\phi^b]_i[\lambda_o]\|^2) \quad (23)$$

The pseudo code of IMM is listed below:

```
[Δφ^b]_0 = [φ_o][X^b]

for i=1, ITMAX
begin;

  [Y]_i = [K_o]^-1 {[F] + [M_o][Δφ^b]_{i-1}⌈λ_o⌋}
  [Z]_i = [Y]_i[R] + [φ_o][ΔR^b]
  [Δφ^b]_i = [Z]_i[α]_i
  if ε_i < TOLERANCE exit loop;
end;
```

Examples

The following example problems were run in Version 66 of MSC/NASTRAN. Both Nelson's method and IMM were implemented in DMAP. A previous DMAP version of Nelson's method[9] was rewritten to take advantage of the new FORTRAN like capabilities in Version 66 DMAP (e.g. SUBDMAPs, if-then constructs, etc.). A perturbation of $\Delta b = 0.01$ was used for each design variable. The maximum allowable error ϵ was 1.0 x 10^{-4} (e.g. $\epsilon \leq 1.0$ x 10^{-4}).

Ten-Bar Truss
The ten-bar truss shown in Fig. 1 and Appendix A demonstrates the use of various boundary conditions. SPCs are at grid 5 and an MPC is at grid 6. The problem has a total of 9 degrees of freedom. It was previously analyzed using Nelson's method in ref. [9]. The material properties are E = 1.0 x 10^7, $\nu = 0.3$, and $\rho = 0.1\frac{lbf}{in^3}$. The planar-truss was modeled using rod elements. The first three modes were selected as constraints. Three design variables were chosen:

- design variable 1: area of element 1
- design variable 2: area of element 5
- design variable 3: Young's modulus

A comparison of $[\Delta\phi^b]$ for Fox's method, Nelson's method, and IMM is shown in Tables 1-10. Only the first 3 modes of the 9 total are used. Table 1 shows the error norm ϵ for IMM. Iteration 0, which is equivalent to Fox's method, has relatively high errors. However, after five iterations the error is less than 1.0 x 10^{-4}. Note that $[\Delta\phi^b]$ with respect to design variable 3 (Young's modulus) should be exactly zero. Tables 2-10 list the eigenvector derivatives for the first three modes with respect to the three design variables. For design variables 1 and 2 best correlation between IMM and Nelson's method is shown for mode 1 (tables 2 and 6). For design variable 3 the derivatives are not exactly zero due to machine roundoff (see tables 8-10).

Although three decompositions were necessary in Nelson's method, the cpu costs for this problem are comparable for both IMM and Nelson's method. Because the size of the problem was so small, the overhead in

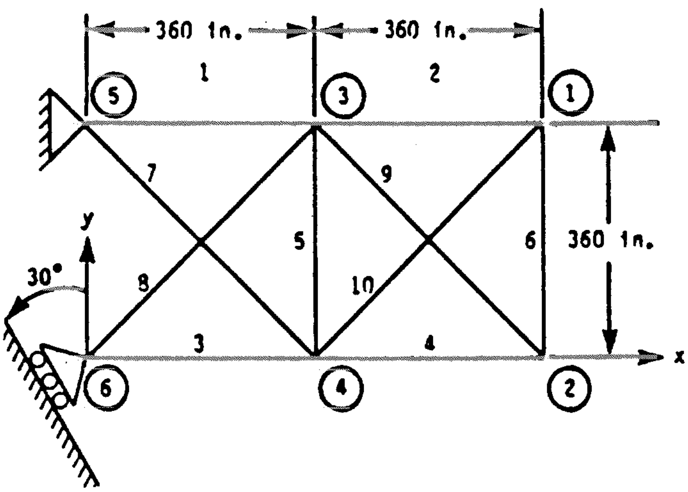

Figure 1: Ten-Bar Truss

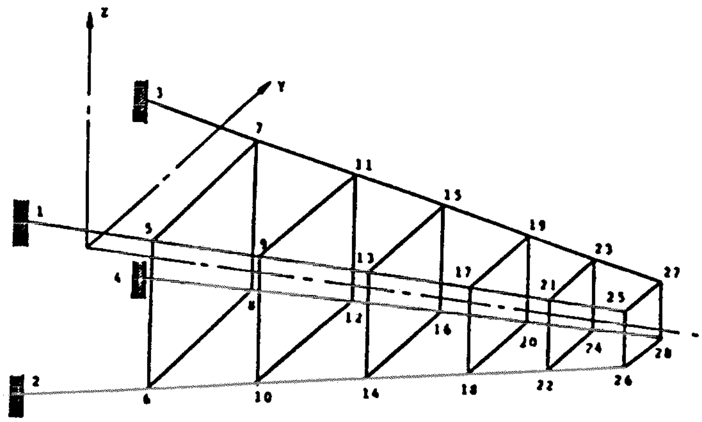

Figure 2: Helicopter Tail Boom

matrix multiplications, etc. in IMM offset decomposition costs in Nelson's method.

Helicopter Tail Boom

The helicopter tail boom shown in Fig. 2 and Appendix B was modeled using 48 beam elements with a total of 144 degrees of freedom. One design variable representing the area of elements one though eight (the eight elements closest to the base) was used. Grids 1-4 were fully constrained. The material properties are $E = 1.0 \times 10^7$, $\nu = 0.3$, and $\rho = 0.1 \frac{lbf}{in^3}$. In the baseline analysis 9 eigenpairs were obtained. Eigenvector derivatives were then calculated for the first nine modes. Table 11 compares the cost and error of IMM (after 10 and 14 iterations) with Nelson's method. Although the size of this problem is still relatively small, a 25-40 % reduction in cpu was observed. After 10 iterations the error norm $\epsilon = 5.5 \times 10^{-2}$ and IMM is 39 percent cheaper than Nelson's method. After 14 iterations, convergence is achieved ($\epsilon \leq 1.0 \times 10^{-4}$) with a cost savings of 26 percent over Nelson's method. It should be noted that iteration 0, which is equivalent to Fox's method, has a very high error $\epsilon = 1.6 \times 10^7$.

Conclusions

The iterative modal method presented here requires that $[K_o]$ be decomposed only once for all eigenvalues and design variables while Nelson's method requires the decomposition of a similar matrix for each eigenvector. For small problems both Nelson's method and IMM will probably be comparable in cost. However, if the size of $[K_o]$ is large and/or many eigenvector derivatives are required, IMM holds promise as being a much cheaper alternative to Nelson's method.

Besides providing an alternative to Nelson's method IMM could also be used as a method for automating reanalysis techniques. Eq. (15) must be rewritten to include the full third-order perturbations. In addition an equation to update the eigenvalues must be included in the iteration scheme. Initial testing has shown the method converges to the overall finite difference answer. Additional testing for larger perturbations and different problem sizes needs to be performed.

References

1. Fox,R.L. and Kapoor,M.P., "Rates of Change of Eigenvalues and Eigenvectors," AIAA Journal, Vol. 6, Dec. 1968, pp. 2426-2429.

2. Rusidill, C.S. and Chu, Y., "Numerical Methods for Evaluating the Derivatives of Eigenvalues and Eigenvectors," AIAA Journal, Vol. 13, June 1975, pp. 834-837.

3. Nelson, R.B., "Simplified Calculation of Eigenvector Derivatives," AIAA Journal, Vol. 14, Sep. 1976, pp. 1201-1205.

4. Herting,D.N., "Eigenvector Derivatives with Second Order Terms," MSC Memo DNH-43, July 6, 1984.

5. Andrew, A.L., "Convergence of an Iterative Method for Derivatives of Eigensystems," Journal of Computational Physics, Vol. 26, 1978, pp. 107-112.

6. Tan, R.C.E., "Accelerating the Convergence of an Iterative Method for Derivatives of Eigensystems," Journal of Computational Physics, Vol. 67, 1986, pp. 230-235.

7. Craig, R.R., Jr., *Structural Dynamics-An Introduction to Computer Methods*, Wiley, New York, 1981.

8. 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.

9. "Calculation of Eigenvector Derivatives in Design Sensitivity Analysis," MSC/NASTRAN Application Manual, January 1986.

	Error Norm ϵ		
Iteration	Des. Var. 1	Des. Var. 2	Des. Var 3
0	2.211×10^{5}	6.932×10^{5}	5.710×10^{-24}
1	9.762×10^{2}	1.480×10^{1}	3.726×10^{-25}
2	7.211×10	4.329×10^{-1}	
2	7.211×10	4.329×10^{-1}	
3	8.210×10^{-2}	1.413×10^{-2}	
4	1.642×10^{-3}	4.778×10^{-4}	
5	4.792×10^{-5}	1.632×10^{-5}	

Table 1: 10-Bar Truss-Error Norms

$\frac{\partial \phi_1}{\partial b_1}$		
Fox	Nelson	IMM
1.12749E-03	2.09119E-03	2.09119E-03
-9.50577E-03	-9.02087E-03	-9.02087E-03
-5.56001E-03	-4.69811E-03	-4.69811E-03
-8.58739E-03	-8.23979E-03	-8.23979E-03
1.12531E-03	2.80090E-03	2.80090E-03
-6.15390E-04	-6.21863E-04	-6.21863E-04
-4.77673E-03	-4.58304E-03	-4.58304E-03
-4.31854E-03	-4.35284E-03	-4.35284E-03
-4.24597E-03	-4.49075E-03	-4.49075E-03
7.35424E-05	7.77821E-03	7.77821E-03

Table 2: 10-Bar Truss-Eigenvector Derivatives

$\frac{\partial \phi_2}{\partial b_1}$		
Fox	Nelson	IMM
-1.09374E-01	-1.43603E-01	-1.43603E-01
1.07970E-01	1.09742E-01	1.09742E-01
3.22870E-02	1.39294E-02	1.39298E-02
1.65853E-01	1.65841E-01	1.65841E-01
-2.59220E-02	-6.97055E-02	-6.97054E-02
7.47223E-02	6.43407E-02	6.43404E-02
1.09142E-02	4.21369E-03	4.21374E-03
-2.12423E-01	-2.11284E-01	-2.11284E-01
4.86056E-03	6.91196E-03	6.91178E-03
-8.41873E-05	-1.19719E-02	-1.19716E-02

Table 3: 10-Bar Truss-Eigenvector Derivatives

$\frac{\partial \phi_3}{\partial b_1}$		
Fox	Nelson	IMM
-1.38140E-02	1.87262E-02	1.87249E-02
1.71786E-01	1.71124E-01	1.71124E-01
2.24565E-02	4.07741E-02	4.07725E-02
1.47434E-01	1.48368E-01	1.48367E-01
-2.19368E-02	1.94012E-02	1.94007E-02
7.98297E-02	8.84480E-02	8.84492E-02
9.68988E-03	1.60198E-02	1.60196E-02
1.74978E-01	1.73917E-01	1.73917E-01
-6.09017E-03	-8.65685E-03	-8.65608E-03
1.05485E-04	1.49941E-02	1.49928E-02

Table 4: 10-Bar Truss-Eigenvector Derivatives

$\frac{\partial \phi_1}{\partial b_2}$		
Fox	Nelson	IMM
-1.41291E-02	-1.20004E-02	-1.20004E-02
3.33138E-03	1.56737E-03	1.56737E-03
5.69656E-03	4.88100E-03	4.88100E-03
1.30416E-02	1.14095E-02	1.14095E-02
-2.00996E-03	-1.78292E-03	-1.78292E-03
2.40253E-03	4.37782E-03	4.37782E-03
3.59275E-03	3.90717E-03	3.90717E-03
-4.37148E-02	-4.37671E-02	-4.37671E-02
4.03189E-03	4.84648E-03	4.84648E-03
-6.98344E-03	-8.39435E-03	-8.39435E-03

Table 5: 10-Bar Truss-Eigenvector Derivatives

$\frac{\partial \phi_2}{\partial b_2}$		
Fox	Nelson	IMM
3.39805E-02	3.25871E-02	3.25871E-02
-3.77131E-02	-3.63893E-02	-3.63894E-02
-2.17873E-02	-2.11549E-02	-2.11550E-02
-5.52999E-02	-5.41765E-02	-5.41766E-02
8.19465E-03	8.06813E-03	8.06811E-03
-1.47565E-02	-1.61233E-02	-1.61232E-02
-1.48057E-02	-1.49985E-02	-1.49986E-02
7.18146E-02	7.18484E-02	7.18484E-02
-1.34000E-02	-1.39829E-02	-1.39829E-02
2.32094E-02	2.42191E-02	2.42190E-02

Table 6: 10-Bar Truss-Eigenvector Derivatives

$\frac{\partial \phi_3}{\partial b_2}$		
Fox	Nelson	IMM
3.65908E-03	-1.06361E-03	-1.06333E-03
-7.47008E-02	-7.03235E-02	-7.03244E-02
-2.08673E-02	-1.87167E-02	-1.87175E-02
-6.28525E-02	-5.91373E-02	-5.91379E-02
8.89753E-03	8.46342E-03	8.46327E-03
-2.35950E-02	-2.81344E-02	-2.81337E-02
-1.55492E-02	-1.61653E-02	-1.61654E-02
-7.22177E-02	-7.21061E-02	-7.21061E-02
-9.48764E-03	-1.14384E-02	-1.14379E-02
1.64331E-02	1.98118E-02	1.98110E-02

Table 7: 10-Bar Truss-Eigenvector Derivatives

$\frac{\partial \phi_1}{\partial b_3}$		
Fox	Nelson	IMM
1.00225E-16	3.34647E-11	1.72171E-16
3.19283E-16	5.96495E-11	1.41254E-16
5.42538E-17	2.60190E-11	-1.01908E-16
3.28173E-16	2.33461E-11	2.10730E-16
-4.97852E-17	4.65332E-12	-5.68921E-18
1.67406E-16	-1.16915E-11	3.33388E-16
2.06487E-17	5.25842E-13	9.86079E-17
8.74442E-17	1.56375E-10	8.33450E-17
-5.89162E-18	-3.41820E-11	7.09810E-17
1.02046E-17	5.92049E-11	-1.22943E-16

Table 8: 10-Bar Truss-Eigenvector Derivatives

$\frac{\partial \phi_2}{\partial b_3}$		
Fox	Nelson	IMM
7.98030E-16	-2.11670E-08	7.01078E-16
-7.40299E-16	6.21412E-08	-8.47776E-16
-3.15912E-16	1.03943E-09	-7.16859E-17
-1.17548E-15	6.38744E-09	-1.32044E-15
1.80860E-16	-1.01782E-09	1.23093E-16
-4.45506E-16	2.58343E-09	-4.68902E-16
-1.62412E-16	1.16770E-10	4.37723E-17
1.69535E-15	-1.06772E-08	1.80954E-15
-1.29384E-16	-1.55032E-10	-6.62955E-17
2.24100E-16	2.68523E-10	1.14827E-16

Table 9: 10-Bar Truss-Eigenvector Derivatives

$\frac{\partial \phi_3}{\partial b_3}$		
Fox	Nelson	IMM
-2.88151E-16	-3.48150E-09	1.15324E-16
2.42164E-15	4.08361E-08	-1.51119E-15
1.93470E-16	5.59429E-09	-2.49847E-16
2.14081E-15	3.52698E-08	-1.49959E-15
-3.24852E-16	-5.23329E-09	5.98766E-17
1.27618E-15	1.88892E-08	-6.08389E-16
2.06130E-18	2.50106E-09	-7.65521E-17
2.30581E-15	3.88781E-08	-1,49906E-15
-2.27614E-16	-1.21708E-09	1.09836E-16
3.94240E-16	2.10805E-09	-1.90241E-16

Table 10: 10-Bar Truss-Eigenvector Derivatives

Iterations	Error Norm ϵ	CPU(MIN)	%Δ Nelson's Method
0	1.648+7		
10	5.498-2	2:58	-39 %
14	8.09-5	3:36	-26 %

Table 11: Tail Boom Model: CPU Comparison with Nelson's Method

APPENDIX A: Listing of Input Data for Example 1

```
ID MSC,TENBAR$
TIME 5
SOL 63
CEND
TITLE      = 10 MEMBER TRUSS
SUBTITLE   = MODAL ANALYSIS
SEALL      = ALL
DISP       = ALL
METHOD     =1
MPC        = 6 $ USED WHEN GRD 6 AT WALL MOVES AT 30 DEG ANGLE
BEGIN BULK
GRID    1       0       720.0   360.0   0.0             3456
GRID    2       0       720.0   0.0     0.0             3456
GRID    3       0       360.0   360.0   0.0             3456
GRID    4       0       360.0   0.0     0.0             3456
GRID    5       0       0.0     360.0   0.0             123456
GRID    6       0       0.0     0.0     0.0             3456
$ BOUNDARY CONDITIONS
SPC     6       6       12
MPC     6       6       1       .86602546       2       .5
$
CROD    1       21      5       3
CROD    2       22      3       1
CROD    3       23      6       4
CROD    4       24      4       2
CROD    5       25      3       4
CROD    6       26      2       1
CROD    7       27      5       4
CROD    8       28      6       3
CROD    9       29      3       2
CROD    10      30      4       1
$
PROD    21      120     28.6
PROD    22      120     0.2
PROD    23      120     23.6
PROD    24      120     15.4
PROD    25      120     0.2
PROD    26      120     0.2
```

```
PROD    27      120     3.0
PROD    28      120     21.0
PROD    29      120     21.8
PROD    30      120     0.2
$
MAT1    120     1.0E7           0.30    0.1
PARAM   WTMASS  .002588
PARAM   COUPMASS1
$ LANCOS, 3 MODES
EIGRL,1,,,3
ENDDATA
```

```
RESTART
$
ID MSC, MODAL $ GDH 6/1/89
SOL LEIGDER $
DIAG 8 $ MATRIX TRAILERS
TIME 5
CEND
$ CASE CONTROL DATA
SENSITY(STORE,FORT) = ALL
TITLE = EIGENVECTOR DERIVATIVES: ITERATIVE MODAL METHOD
SUBTITLE = 3 CONSTRAINTS, 3 DESIGN VARIABLES
$
BEGIN BULK
PARAM,WTMASS,.002588
PARAM,COUPMASS,1 $
PARAM,ITERATE,YES $ PERFORM ITERATIVE MODAL METHOD
PARAM,ITRPRNT,YES $ PRINT AT EACH ITERATION
PARAM,ITMAX,10    $ DO A MAXIMUM OF 10 ITERATIONS
PARAM,TOL,1.E-4   $ CONVERGENCE TOLERANCE
$ 1ST 3 MODES AS CONSTRAINTS
DSCONS  51      L1MIN   LAMA    1                       0.00    MIN
DSCONS  52      L2MIN   LAMA    2                       0.00    MIN
DSCONS  53      L3MIN   LAMA    3                       0.00    MIN
$ 3 DESIGN VARIABLES
DVAR    1       AREA1   0.01    41
DVAR    5       AREA5   0.01    45
DVAR    99      YMOD    0.01    199
DVSET   41      PROD    4               1.0     21
DVSET   45      PROD    4               1.0     25
DVSET   199     PROD    3       91              21 THRU 30
MAT1    120     1.E7                    0.1     12.9-6
MAT1    91      1.01E7                  0.1     12.9-6
PARAM   DSZERO  1.E-2
ENDDATA
```

APPENDIX B: Listing of Input Data for Example 2

```
ID MSC, MODE9 $
TIME 5
SOL 63
CEND
TITLE     = 48 MEMBER HELICOPTER BOOM
SUBTITLE  = MODAL ANALYSIS
SEALL     = ALL
DISP      = ALL
METHOD    =1
BEGIN BULK
GRID,1,,0.0,-12., 10.,,123456
GRID,2,,0.0,-12.,-10.,,123456
GRID,3,,0.0, 12., 10.,,123456
GRID,4,,0.0, 12.,-10.,,123456
$
GRID,5,,10.0,-10.5, 9.0
GRID,6,,10.0,-10.5,-9.0
GRID,7,,10.0, 10.5, 9.0
GRID,8,,10.0, 10.5,-9.0
$
GRID,9 ,,20.0,-9.0 , 8.0
GRID,10,,20.0,-9.0 ,-8.0
GRID,11,,20.0, 9.0 , 8.0
GRID,12,,20.0, 9.0 ,-8.0
$
GRID,13,,30.0,-7.5 , 7.0
GRID,14,,30.0,-7.5 ,-7.0
GRID,15,,30.0, 7.5 , 7.0
GRID,16,,30.0, 7.5 ,-7.0
$
GRID,17,,40.0,-6.0 , 6.0
GRID,18,,40.0,-6.0 ,-6.0
GRID,19,,40.0, 6.0 , 6.0
GRID,20,,40.0, 6.0 ,-6.0
$
GRID,21,,50.0,-4.5 , 5.0
GRID,22,,50.0,-4.5 ,-5.0
GRID,23,,50.0, 4.5 , 5.0
```

```
GRID,24,,50.0, 4.5 ,-5.0
$
GRID,25,,60.0,-3.0 , 4.0
GRID,26,,60.0,-3.0 ,-4.0
GRID,27,,60.0, 3.0 , 4.0
GRID,28,,60.0, 3.0 ,-4.0
$
CBAR,1,1,1,5,0.,0.,1.
CBAR,2,1,2,6,0.,0.,1.
CBAR,3,1,3,7,0.,0.,1.
CBAR,4,1,4,8,0.,0.,1.
$
CBAR,5,2,6,5,1.,0.,0.
CBAR,6,2,8,7,1.,0.,0.
CBAR,7,2,5,7,1.,0.,0.
CBAR,8,2,6,8,1.,0.,0.
$
CBAR,9,3,5,9,0.,0.,1.
CBAR,10,3,6,10,0.,0.,1.
CBAR,11,3,7,11,0.,0.,1.
CBAR,12,3,8,12,0.,0.,1.
$
CBAR,13,4,10,9,1.,0.,0.
CBAR,14,4,12,11,1.,0.,0.
CBAR,15,4,9,11,1.,0.,0.
CBAR,16,4,10,12,1.,0.,0.
$
CBAR,17,5,9,13,0.,0.,1.
CBAR,18,5,10,14,0.,0.,1.
CBAR,19,5,11,15,0.,0.,1.
CBAR,20,5,12,16,0.,0.,1.
$
CBAR,21,6,14,13,1.,0.,0.
CBAR,22,6,16,15,1.,0.,0.
CBAR,23,6,13,15,1.,0.,0.
CBAR,24,6,14,16,1.,0.,0.
$
CBAR,25,7,13,17,0.,0.,1.
CBAR,26,7,14,18,0.,0.,1.
CBAR,27,7,15,19,0.,0.,1.
CBAR,28,7,16,20,0.,0.,1.
```

```
$
CBAR,29,8,18,17,1.,0.,0.
CBAR,30,8,20,19,1.,0.,0.
CBAR,31,8,17,19,1.,0.,0.
CBAR,32,8,18,20,1.,0.,0.
$
CBAR,33,9,17,21,0.,0.,1.
CBAR,34,9,18,22,0.,0.,1.
CBAR,35,9,19,23,0.,0.,1.
CBAR,36,9,20,24,0.,0.,1.
$
CBAR,37,10,22,21,1.,0.,0.
CBAR,38,10,24,23,1.,0.,0.
CBAR,39,10,21,23,1.,0.,0.
CBAR,40,10,22,24,1.,0.,0.
$
CBAR,41,11,21,25,0.,0.,1.
CBAR,42,11,22,26,0.,0.,1.
CBAR,43,11,23,27,0.,0.,1.
CBAR,44,11,24,28,0.,0.,1.
$
CBAR,45,12,26,25,1.,0.,0.
CBAR,46,12,28,27,1.,0.,0.
CBAR,47,12,25,27,1.,0.,0.
CBAR,48,12,26,28,1.,0.,0.
$
PBAR,1 ,1,.05,.05,.05,.05
PBAR,2 ,1,.05,.05,.05,.05
PBAR,3 ,1,.05,.05,.05,.05
PBAR,4 ,1,.05,.05,.05,.05
PBAR,5 ,1,.05,.05,.05,.05
PBAR,6 ,1,.05,.05,.05,.05
PBAR,7 ,1,.05,.05,.05,.05
PBAR,8 ,1,.05,.05,.05,.05
PBAR,9 ,1,.05,.05,.05,.05
PBAR,10,1,.05,.05,.05,.05
PBAR,11,1,.05,.05,.05,.05
PBAR,12,1,.05,.05,.05,.05
MAT1,1,1.E7,,.30,.10
PARAM,COUPMASS,1
PARAM,WTMASS,.002588
```

```
PARAM,GRDPNT,1
$ LANCOS, 9 MODES
EIGRL,1,0.0,,9
ENDDATA
```

```
$
RESTART $
ID MSC, MODAL $ GDH 6/2/89
SOL LEIGDER $
DIAG 8 $ MATRIX TRAILERS
TIME 5
CEND
$ CASE CONTROL DATA
SENSITY(STORE,FORT) = ALL
TITLE = EIGENVECTOR DERIVATIVES: ITERATIVE MODAL METHOD
SUBTITLE = 9 CONSTRAINTS, 1 DESIGN VARIABLE
$
BEGIN BULK
PARAM,WTMASS,.002588
PARAM,COUPMASS,1   $
PARAM,ITERATE,YES  $ PERFORM ITERATIVE MODAL METHOD
PARAM,ITRPRNT,YES  $ PRINT AT EACH ITERATION
PARAM,ITMAX,20     $ DO A MAXIMUM OF 20 ITERATIONS
PARAM,TOL,1.E-4    $ CONVERGENCE TOLERANCE
$ 1ST 9 MODES AS CONSTRAINTS
DSCONS  51      L1MIN   LAMA    1                       0.00    MIN
DSCONS  52      L2MIN   LAMA    2                       0.00    MIN
DSCONS  53      L3MIN   LAMA    3                       0.00    MIN
DSCONS  54      L4MIN   LAMA    4                       0.00    MIN
DSCONS  55      L5MIN   LAMA    5                       0.00    MIN
DSCONS  56      L6MIN   LAMA    6                       0.00    MIN
DSCONS  57      L7MIN   LAMA    7                       0.00    MIN
DSCONS  58      L8MIN   LAMA    8                       0.00    MIN
DSCONS  59      L9MIN   LAMA    9                       0.00    MIN
$ DESIGN VARIABLE: AREA OF PBAR 1 & 2
DVAR    1       VAR1    0.01    41
DVSET   41      PBAR    4               1.0     1       2
PARAM   DSZERO  1.E-2
ENDDATA
```