Eigenvalue Routines in NASTRAN*

A Comparison with the Block Lanczos Method

N94-17838

S1-39
190531
P.33

by

V. A. Tischler and V. B. Venkayya
Wright Laboratory
Wright Patterson AFB OH 45433-6553

SUMMARY

The NASA STRuctural ANalysis (NASTRAN) (Ref 1) program is one of the most extensively used engineering applications software in the world. It contains a wealth of matrix operations and numerical solution techniques, and they were used to construct efficient eigenvalue routines. The purpose of this paper is to examine the current eigenvalue routines in NASTRAN and to make efficiency comparisons with a more recent implementation of the Block Lanczos algorithm by Boeing Computer Services (BCS). This eigenvalue routine is now available in the BCS mathematics library as well as in several commercial versions of NASTRAN. In addition, CRAY maintains a modified version of this routine on their network. Several example problems, with a varying number of degrees of freedom, were selected primarily for efficiency bench-marking. Accuracy is not an issue, because they all gave comparable results. The Block Lanczos algorithm was found to be extremely efficient, in particular, for very large size problems.

INTRODUCTION

In NASTRAN the real eigenvalue analysis module is used to obtain structural vibration modes from the symmetric mass and stiffness matrices, M_{AA} and K_{AA}, which are generated in the program using finite element models. Currently the user has a choice of four methods for solving vibration mode problems: Determinant Method, Inverse Power Method with Shifts, Tridiagonal Method (Givens' Method) and Tridiagonal Reduction or FEER Method. NASTRAN provides all these options for user convenience as well as for analysis efficiency. For example, the Givens' Method is most appropriate when all the eigenvalues are of equal interest. By the same token, it is not suitable (because of the need for excessive computer resources) when the number of degrees of freedom is too large (greater than three to four hundred) unless preceded by Guyan reduction (ASET or OMIT). The Inverse Power, Determinant and FEER Methods are most suitable when only a small subset of the eigenvalues are of interest. These methods take advantage of the sparseness of the mass and stiffness matrices and extract one or a small subset of eigenvalues at a time.

---

*NASTRAN without qualification refers to COSMIC-NASTRAN (or government version) in the paper.

The purpose of this paper is to examine, in some detail, the real eigenvalue analysis methods currently available in NASTRAN and to make efficiency comparisons with the Block Lanczos algorithm as implemented by Boeing Computer Services (BCS) and currently available in some commercial versions of NASTRAN (for example MSC-NASTRAN and UAI-NASTRAN). The accuracy of the eigenvalues is not an issue in this paper, because all the methods gave comparable results. Efficiency in terms of computer time is the only issue in this bench-marking. This study was made, for all cases, on a single platform, the CRAY XMP. The genesis of the Block Lanczos Method in all the NASTRANs, as well as the CRAY version, is the one implemented by BCS with some modifications.

Section 1 discusses the general form of the eigenvalue problem for vibration modes. In Section 2 a mathematical formulation of the four methods in NASTRAN is given with emphasis on the FEER Method as a precursor to the Lanczos Method. A detailed mathematical description of the Block Lanczos Method is given in Section 3. Also reference is made to the Lanczos method in MSC NASTRAN and to its implementation by CRAY Research, Inc. In Section 4 selected frequencies are calculated for five structures of varying complexity using the Inverse Power Method, the FEER Method, MSC/NASTRAN Lanczos Method and CRAY Lanczos Method. Results are discussed in Section 5 and recommendations are made for possible implementation into NASTRAN.

1.0 The Eigenvalue Problem

1.1 The general form of the eigenvalue problem for vibration modes is

$$Kx = \lambda Mx \tag{1}$$

where M and K are the symmetric mass and stiffness matrices, the eigenvalue $\lambda = \omega^2$ the square of the natural vibration frequency, and x is the eigenvector corresponding to λ. The dimension of the matrices K and M is nxn, where n is the number of degrees of freedom in the analysis set. For this paper it is assumed that K and M are at least positive semi-definite. Thus associated with Eq (1) are n eigenpairs λ_i, x_i such that

$$Kx_i = \lambda_i Mx_i \qquad i = 1, 2, \ldots, n \tag{2}$$

Properties of the eigenvectors include:

$$x_i^T Mx_j = \begin{cases} M_{ii} \text{ for } & i = j \\ 0 \text{ for } & i \neq j \end{cases} \tag{3}$$

where M_{ii} is referred to as the modal mass or generalized mass. It is evident from Eq (3) that the eigenvectors are orthonormal with respect to the mass matrix. Also the eigenvectors are orthonormal with respect to the stiffness matrix, i.e.

$$x_i^T Kx_j = \begin{cases} K_{ii} \text{ for } & i = j \\ 0 \text{ for } & i \neq j \end{cases} \tag{4}$$

where K_{ii} is the modal stiffness or generalized stiffness.

The Rayleigh quotient shows that the modal mass, M_{ii}, and modal stiffness, K_{ii}, are related to the eigenvalue λ_i, i.e.

$$\lambda_i = \frac{x_i^T Kx_i}{x_i^T Mx_i} = \frac{K_{ii}}{M_{ii}} \tag{5}$$

For normalized eigenvectors with respect to modal mass, Eqs (3) can be written as

$$x_i^T Mx_j = \begin{cases} 1 \text{ for } & i = j \\ 0 \text{ for } & i \neq j \end{cases} \tag{6}$$

Now using Eqs (5), Eqs (4) can be written as

$$x_i^T Kx_j = \begin{cases} \lambda_j \text{ for } & i = j \\ 0 \text{ for } & i \neq j \end{cases} \tag{7}$$

The central issue of a real eigenvalue or normal modes analysis is to determine the eigenvalues, λ_i, and the eigenvectors, x_i, which satisfy the conditions stated by Eqs (1-7). The next sections present the important elements of the eigenvalue methods of interest.

2.0 Eigenvalue Extraction Methods in NASTRAN

2.1 For real symmetric matrices there are four methods of eigenvalue extraction available in NASTRAN: the Determinant Method, the Inverse Power Method with shifts, the Givens' Method of Tridiagonalization and the Tridiagonal Reduction or FEER Method. Most methods of algebraic eigenvalue extraction can be categorized as belonging to one or the other of two groups: transformation methods and tracking methods. In a transformation method the two matrices M and K are simultaneously subjected to a series of transformations with the object of reducing them to a special form (diagonal or triadiagonal) from which eigenvalues can be easily extracted. These transformations involve pre and post multiplication by elementary matrices to annihilate the off-diagonal elements in the two matrices. This process preserves the original eigenvalues in tact in the transformed matrices. The ratio of the diagonal elements in the two matrices gives the eigenvalues. In a tracking method the roots are extracted, one at a time, by iterative procedures applied to the dynamic matrix consisting of the original mass and stiffness matrices. In NASTRAN the Givens' and the FEER methods are transformation methods, while the Determinant and the Inverse Power methods are tracking methods. Both tracking methods and the Givens' method will be discussed briefly in this section while the Lanczos algorithm, the main emphasis of this paper, is outlined here and in more detail in the next section.

2.2 Determinant Method

For the vibration problem

$$Kx = \lambda Mx \tag{8}$$

the matrix of coefficients, A, has the form

$$A = K - \lambda M \tag{9}$$

The determinant of A can be expressed as a function of λ, i.e.

$$D(A) = |A| = (\lambda - \lambda_1)(\lambda - \lambda_2)\dots(\lambda - \lambda_n)$$

where λ_i, $i = 1, 2 \dots n$ are the eigenvalues of A. In the determinant method $D(A)$ is evaluated for trial values of λ, selected according to an iterative procedure, and a criterion is established to determine when $D(A)$ is sufficiently small or when λ is sufficiently close to an eigenvalue. The procedure used for evaluating $D(A)$ employs the triangular decomposition

$$A = LU \tag{10}$$

for an assumed value of λ where L is a lower unit triangular matrix and U is an upper triangular matrix. $D(A)$ is equal to the product of the diagonal terms of U. Once an approximate eigenvalue, λ_i, has been accepted, an eigenvector, x_i, is determined from

$$LUx_i = 0 \tag{11}$$

by back substitution where one of the elements of x_i is preset. Since $L(\lambda_i)$ is nonsingular, only $U(\lambda_i)$ is needed. The determinant method may not be efficient in some cases if more than a few eigenvalues are desired because of the large number of triangular decompositions of A.

2.3 Inverse Power Method with Shifts

The Inverse Power Method with shifts is an iterative procedure applied directly to Eq (1) in the form

$$[K - \lambda M]\,x = 0 \tag{12}$$

It is required to find all the eigenvalues and eigenvectors within a specified range of λ. Let

$$\lambda = \lambda_o + \Lambda \tag{13}$$

where λ_o is a constant called the shift point. Therefore Λ replaces λ as the eigenvalue. The iteration algorithm is defined in the *nth* iteration step by:

$$[K - \lambda_o M]\,w_n = Mx_{n-1} \tag{14}$$

$$x_n = \frac{1}{c_n} w_n \tag{15}$$

where c_n, a scaler, is equal to that element of the vector w_n with the largest absolute value. At convergence $1/c_n$ converges to Λ, the shifted eigenvalue closest to the shift point, and x_n converges to the corresponding eigenvector ϕ_i. Note from Eq (14) that a triangular decomposition of matrix $K - \lambda M$ is necessary in order to evaluate w_n. The shift point λ_o can be changed in order to improve the rate of convergence toward a particular eigenvalue or to improve accuracy and convergence rates after several roots have been extracted from a given shift point. Also λ_o can be calculated such that the eigenvalues within a desired frequency band can be found and not just those that have the smallest absolute value.

For calculating additional eigenvalues, the trial vectors, x_n, in Eq (14) must be swept to eliminate contributions due to previously found eigenvalues that are closer to the shift point than the current eigenvalue. An algorithm to accomplish this is given as follows:

$$x_n = \bar{x}_n - \sum_{i=1}^{m} (\bar{\phi}_i^t M \bar{x}_n)\,\bar{\phi}_i \tag{16}$$

where $\bar{x}_n$ is the trial vector being swept, m is the number of previously extracted eigenvalues, and $\bar{\phi}_i$ is defined by

$$\bar{\phi}_i = \frac{x_{i,N}}{\sqrt{x_{i,N}^T M x_{i,N}}} \tag{17}$$

where $x_{i,N}$ is the last eigenvector found in iterating for the *ith* eigenvalue.

The inverse power method allows the user to define a range of interest $[\lambda_a, \lambda_b]$ on the total frequency spectrum and to request a desired number of eigenvalues, ND, within that range. When ND is greater than the actual number of eigenvalues in the range, then the method guarantees the lowest eigenvalues in the range.

2.4 Givens' Method of Tridiagonalization

In the Givens' method the vibration problem as posed by Eq (8) is first transformed to the form

$$Ax = \lambda x \tag{18}$$

by the following procedures. The mass matrix, M, is decomposed into upper and lower triangular matrices such that

$$M = LL^T \tag{19}$$

If M is not positive definite, the decomposition in Eq (19) is not possible. For example, when a lumped mass model is used, NASTRAN does not compute rotary inertia effects. This means that the rows and columns of the mass matrix corresponding to the rotational degrees of freedom are zero resulting in a singular mass matrix. In this case the mass matrix must be modified to eliminate the massless degrees of freedom.

Thus Eq (8) becomes

$$Kx = \lambda LL^T x \tag{20}$$

which implies after premulitplying by L^{-1} and post multiplying by $(L^T)^{-1}$ that

$$L^{-1}K(L^T)^{-1}x = \lambda x \tag{21}$$

i.e.

$$Ax = \lambda x$$

where $A=L^{-1}K(L^T)^{-1}$. Note that L^{-1} is easy to perform, since L is triangular. Also $A = L^{-1}K(L^T)^{-1}$ is a symmetric matrix. The matrix A is then transformed to a tridiagonal matrix, A_r, by the Givens' method, i.e a sequence of orthogonal transformations, T_j, are defined such that

$$T_r T_{r-1} \dots T_2 T_1 A x = \lambda T_r T_{r-1} \dots T_2 T_1 x \tag{22}$$

Recall that an orthogonal transformation is one whose matrix T satisfies

$$TT^T = T^T T = I \tag{23}$$

the identity matrix. The eigenvalues of A are preserved by the transformation, and if

$$x = T_1^T T_2^T \dots T_{r-1}^T T_r^T y \tag{24}$$

then from Eq (22)

$$T_r T_{r-1} \dots T_2 T_1 A T_1^T T_2^T \dots T_{r-1}^T T_r^T y = \lambda T_r T_{r-1} \dots T_2 T_1 T_1^T T_2^1 \dots T_r^T y$$

i.e.

$$T_r T_{r-1} \dots T_2 T_1 A T_1^T T_2^T \dots T_{r-1}^T T_r^T y = \lambda y \tag{25}$$

by repeatedly applying Eq (23). Eq (25) implies that y is an eigenvector of the transformed matrix $T_r T_{r-1} \dots T_2 T_1 A T_1^T T_2^T \dots T_{r-1}^T T_r^T$. Thus x can be obtained from y by Eq (24).

The eigenvalues of the tridiagonal matrix, A_r, are extracted using a modified Q-R algorithm, i.e., $A_{r+1} = Q_r^T A_r Q_r$ such that A_r is factored into the product $Q_r R_r$ where R_r is an upper triangular matrix and Q_r is orthogonal. Thus

$$A_r = Q_r R_r \tag{26}$$

and

$$\begin{aligned} A_{r+1} &= Q_r^T A_r Q_r \\ &= Q_r^T Q_r R_r Q_r \qquad \text{from Eq (26)} \end{aligned}$$

Since Q_r is orthogonal, then

$$A_{r+1} = R_r Q_r \tag{27}$$

In the limit as $r \rightarrow \infty$ and A is symmetric, A_r will approach a diagonal matrix. Since eigenvalues are preserved under an orthogonal transformation, the diagonal elements of the limiting diagonal matrix will be the eigenvalues of the original matrix A.

To obtain the ith eigenvector, y_i, of the tridiagonal matrix, A_r, the tridiagonal matrix $A_r - \lambda_i I$ is factored such that

$$A_r - \lambda_i I = L_i U_i \tag{28}$$

where L_i is a unit triangular matrix and U_i is an upper triangular matrix. The eigenvector y_i is then obtained by iterating on

$$U_i y_i^{(n)} = y_i^{(n-1)} \tag{29}$$

where the elements of the vector $y_i^{(o)}$ are arbitrary. Note that the solution of Eq (29) is easily obtained by back substitution since U_i has the form

$$U_i = \begin{bmatrix} p_1 \; q_1 \; r_1 & & & \\ & p_2 \; q_2 \; r_2 & & \\ & - \; - \; - & & \\ & - \; - \; - & p_{n-1} & q_{n-1} \\ & & & p_n \end{bmatrix} \tag{30}$$

The eigenvectors of the original coefficient matrix, A, are then obtained from Eq (24).

Note that in the Givens' method the dimension of A equals the dimension of A_r. The major share of the total effort expended in this method is in converting A to A_r. Therefore the total effort is not strongly dependent on the number of eigenvalues extracted.

2.5 Tridiagonal Reduction or FEER Method

The tridiagonal Reduction or FEER method is a matrix reduction scheme whereby the eigenvalues in the neighborhood of a specified point, λ_o, in the eigenspectrum can be accurately determined from a tridiagonal eigenvalue problem whose dimension or order is much lower than that of the full problem. The order of the reduced problem, m, is never greater than

$$m = 2\bar{q} + 10$$

where $\bar{q}$ is the desired number of eigenvalues. So the power of the FEER method lies in the fact that the size of the reduced problem is the same order of magnitude as the number of desired roots, even though the actual finite element model may have thousands of degrees of freedom.

There are five basic step in the FEER method:

1. Eq (8) is converted to a symmetric inverse form

$$Bx = \Lambda Mx \tag{31}$$

where

$$\Lambda = \frac{1}{\lambda - \lambda_o} \tag{32}$$

and λ_o is a shift value.

2. The tridiagonal reduction algorithm or Lanczos algorithm is used to transform Eq (31) into a tridiagonal form of reduced order.

3. The eigenvalues of the reduced matrix are extracted using a Q-R algorithm similar to that described in Section 2.4.

4. Upper and lower bounds on the extracted eigenvalues are obtained.

5. The corresponding eigenvectors are computed and converted to physical form.

To implement Step 1, consider Eq (8),

$$Kx = \lambda Mx$$

When vibration modes are requested in the neighborhood of a specified frequency, λ_o, Eq (8) can be written

$$Kx - \lambda_o Mx = \lambda Mx - \lambda_o Mx$$

$$(K - \lambda_o M)\,x = (\lambda - \lambda_o)\,Mx \tag{33}$$

Let $\bar{K} = K - \lambda_o M$ and $\lambda' = \lambda - \lambda_o$. Then from Eq (33)

$$\bar{K}x = \lambda' Mx \tag{34}$$

$$x = \lambda' \bar{K}^{-1} Mx$$

$$Mx = \lambda' M \bar{K}^{-1} Mx$$

$$M\bar{K}^{-1} Mx = \frac{1}{\lambda'} Mx \tag{35}$$

Factor $\bar{K}$ by Cholesky decomposition, i.e.

$$\bar{K} = Ld'L^T \tag{36}$$

where L is a lower triangular matrix and d' is a diagonal matrix. Then Eq (35) can be written

$$M\left[(L^T)^{-1} d'^{-1} L^{-1} \right] Mx = \frac{1}{\lambda'} Mx$$

i.e.

$$Bx = \Lambda M x$$

where $B = M\left[(L^T)^{-1} d^{-1} L^{-1} \right] M$ and $\Lambda = \frac{1}{\lambda'} = \frac{1}{\lambda - \lambda_o}$. Now step 1 is complete.

To implement Step 2 rewrite Eq (31) as

$$\bar{B}x = \Lambda x$$

where $\bar{B} = M^{-1}B$. Now $\bar{B}$ is reduced to tridiagonal form, A, using single vector Lanczos recurrence formulas defined by

$$\left.\begin{aligned} a_{i,i} &= V_i^T B V_i \\ \bar{V}_{i+1} &= \bar{B}V_i - a_{i,i}V_i - d_i V_{i-1} \\ d_{i+1} &= \{\bar{V}_{i+1}^T M \bar{V}_{i+1}\}^{1/2} \end{aligned}\right\} i = 1, 2, \ldots, m \tag{37}$$

$$V_{i+1} = \frac{1}{d_{i+1}} \bar{V}_{i+1} \qquad i = 1, 2, \ldots m-1$$

where vector V_o=0, V_1 is a random starting vector and d_1=0. The reduced tridiagonal eigenvalue problem is now given as

$$Ay = \begin{bmatrix} a_{11} & d_2 & & & & \\ d_2 & a_{22} & d_3 & & & \\ & d_3 & a_{33} & d_4 & & \\ & & \ddots & \ddots & \ddots & \\ & & & d_{m-1} & a_{m-1,m-1} & d_m \\ & & & & d_m & a_{mm} \end{bmatrix} y = \bar{\Lambda} y \tag{38}$$

where $\bar{\Lambda}$ approximates the eigenvalue Λ of Eq (31), and y is an eigenvector of A. The Lanczos formulas generate a V matrix, vector by vector, i.e.

$$V = [V_1, V_2, \ldots V_m] \tag{39}$$

and Eqs (37) are modified by NASTRAN such that each vector V_{i+1} is re-orthogonalized to all previously computed V vectors, i.e. V is orthonormal to M.

$$V^T M V = I \tag{40}$$

Thus

$$A = V^T B V \tag{41}$$

Note from Eq (41) that A is an mxm matrix.

For step 3 the eigenvalues, $\bar{\Lambda}$, and eigenvectors, y, of Eq (38) are obtained as described for the Givens' method in Section 2.4. The eigenvectors are normalized so that

$$y_i^T y_i = 1 \qquad i = 1, \ldots, m \tag{42}$$

For step 4 the following error bound formula has been derived and serves as a criterion for selecting acceptable eigensolutions

$$\varepsilon_i = \left| 1 - \frac{\bar{\lambda}_i}{\bar{\lambda}_i} \right| \le \left| \frac{d_{m+1} \cdot y_{mi}}{\bar{\Lambda}_i (1 + \lambda_o \bar{\Lambda}_i)} \right| \tag{43}$$

In Eq (43) λ_i is an approximation to the exact eigenvalue λ_i in Eq (8), d_{m+1} is calculated from Eqs (37), y_{mi} is the last component of the mth eigenvector, y_m, of A, and $\bar{\Lambda}_i$ is the ith eigenvalue of A. The ith eigenvalue $\bar{\lambda}_i$ is acceptable, if ε_i is less than or equal to a preset error tolerance.

Now step 5 is implemented for acceptable eigenvalues. If $(\bar{\Lambda}, y)$ is an eigenpair of Eq (38), then

$$Ay = \bar{\Lambda} y$$

or from Eqs (40) and (41)

$$V^T B V y = \bar{\Lambda} V^T M V y$$

$$B V y = \bar{\Lambda} M V y \tag{44}$$

Now if $x=Vy$, then

$$Bx = \bar{\Lambda} M x$$

i.e. $(\bar{\Lambda}, x)$ is an eigenpair of Eq (31).

Thus for step 5 the eigenvectors of Eq (31) or equivalently Eq (8) are calculated from

$$x = Vy \tag{45}$$

and the eigenvalue $\bar{\lambda}$ is calculated from Eq (32) i.e.

$$\bar{\lambda} = \frac{1}{\bar{\Lambda}} + \lambda_o \tag{46}$$

Note that in the FEER method the matrix B enters the recurrence formulas, Eqs (37), only through the matrix-vector multiply terms BV_i. Therefore B is not modified by the computations. Lanczos procedures for real symmetric matrices require only that a user provide a subroutine which for any given vector, z, computes Bz.

3.0 Block Lanczos Method

3.1 Recall that the eigenvalue problem in vibration analysis is given by Eq (8), i.e.

$$Kx = \lambda Mx$$

where K and M are symmetric positive definite matrices. Generally the eigenvalues of interest are the smallest ones, but they are often poorly separated. However, the largest eigenvalues which are not interesting have good separation. Also convergence rates are very slow at the low end of the spectrum and fast at the higher end. Convergence rates can be accelerated to the desired set of eigenvalues by a spectral transformation, i.e. consider the problem

$$M(K - \sigma M)^{-1} Mx = uMx \tag{47}$$

where σ, the shift, is a real parameter. It can be shown that (λ, x) is an eigenpair of Eq (8) if and only if $(\frac{1}{\lambda - \sigma}, x)$ is an eigenpair of Eq (47). The spectral transformation does not change the eigenvectors, but the eigenvalues of Eq (47) are related to the eigenvalues of Eq (8) by

$$u = \frac{1}{\lambda - \sigma} \tag{48}$$

This transformation will allow the Lanczos algorithm to be applied even when M is semidefinite.

Consider the effect of the spectral transformation on a satellite problem which will be discussed in detail in Section 4. Figure 1 shows the shape of the transformation. Table A shows the effect of the transformation using an initial shift of $\sigma = .046037$. Note that the smallest 8 eigenvalues are transformed from closely spaced eigenvalues to eigenvalues with good separation.

Satellite Problem

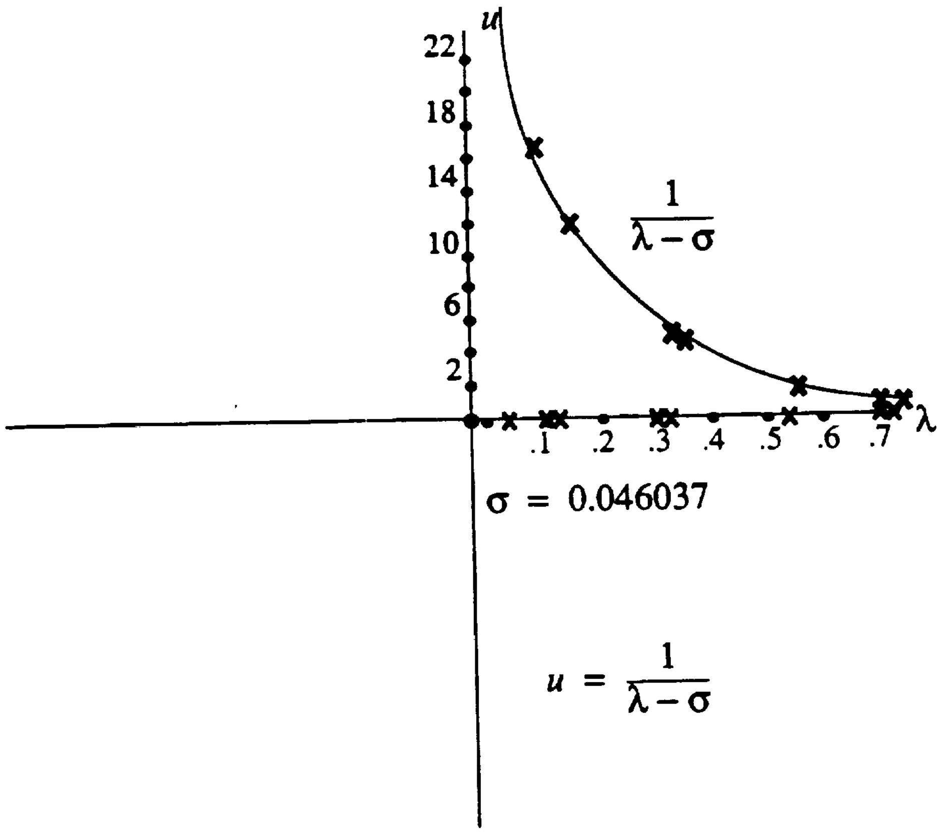

FIGURE 1

	ORIGINAL			TRANSFORMED		
I	$\lambda(i)$	$u(i)$	gap	rel gap	gap	rel gap
1	.07229	38.09088	.03611	.05357	22.05574	.60158
2	.10840	16.03514	.01716	.02546	3.46017	.09438
3	.12556	12.57497	.18740	.27800	8.82857	.240803
4	.31296	3.74640	6.000×10^{-5}	8.9006×10^{-5}	.00084	2.29114×10^{-5}
5	.31302	3.74556	.27055	.40134	1.88521	.05142
6	.58357	1.86035	.16180	.24002	.43042	.01174
7	.74537	1.42993	.00103	.00153	.00210	5.72784×10^{-5}
8	.74640	1.42783				

Table A

Our objective is to define the Spectral Transformation Block Lanczos algorithm. Let's consider first the Basic Block Lanczos Algorithm.

3.2 Basic Block Lanczos Algorithm

Consider the Lanczos Algorithm (Refs 2.3) for the eigenvalue problem.

$$Hx = \lambda x \tag{49}$$

where H is symmetric

The block Lanczos iteration with block size p for an nxn matrix H is given as:

Initialization:

set $Q_o = 0$

set $B_1 = 0$

choose R_1 and orthonormalize the columns of R_1 to obtain Q_1

Lanczos Loop:

For $j = 1, 2, 3, \ldots$

set $U_j = HQ_j - Q_{j-1} B_j^T$

set $A_j = Q_j^T U_j$

set $R_{j+1} = U_j - Q_j A_j$

Compute the orthogonal factorization $Q_{j+1}B_{j+1} = R_{j+1}$

End Loop

Matrices Q_j, U_j, and R_j for $j= 1, 2, \ldots$ are nxp; A_j and B_j are pxp. A_j is symmetric and B_j is upper triangular. The blocksize p is the number of column vectors of Q_j. So if $p = 1$, then Q_j is a column vector, q. Thus the matrix H is not explicitly required, but only a subroutine that computes Hq for a given vector q. A_j and B_j are generalizations of the scalers a_j and d_j in the ordinary Lanczos recurrence.

The recurrence formula in the Lanczos loop can also be written as

$$R_{j+1} = Q_{j+1}B_{j+1} = HQ_j - Q_jA_j - Q_{j-1}B_j^T \tag{50}$$

The orthogonal factorization of the residual, R_{j+1}, implies that the columns of Q_j are orthonormal. Indeed it has been shown that the combined column vectors of the matrices, Q_1, Q_2, ... Q_j, called the Lanczos vectors, form an orthonormal set.

The blocks of Lanczos vectors form an $nxjp$ matrix W_j where

$$W_j = [Q_1, Q_2, \ldots Q_j] \tag{51}$$

From the algorithm itself a *jpxjp* block tridiagonal matrix, T_j, is defined such that

$$T_j = \begin{bmatrix} A_1 & B_2^T & 0 & \dots & 0 \\ B_2 & A_2 & B_3^T & \dots & 0 \\ & & & & \\ 0 & \dots & B_{j-1} & A_{j-1} & B_j^T \\ 0 & \dots & 0 & B_j & A_j \end{bmatrix} \tag{52}$$

Since the matrices B_j are upper triangular, T_j is a band matrix with half band width $p+1$. The first j formulas defined by Eq (50) can be combined using Eqs (51) and (52) into a single formula

$$HW_j = W_jT_j + Q_{j+1}B_{j+1}E_j^T \tag{53}$$

where E_j is an *nxp* matrix of zeros except the last *pxp* block is a pxp identity matrix.

Premulitplying Eq (53) by W_j^T implies

$$W_j^THW_j = W_j^TW_jT_j + W_j^TQ_{j+1}B_{j+1}E_j^T$$

i.e.

$$W_j^THW_j = T_j \tag{54}$$

since

$$W_j^TW_j = I \quad \text{and} \quad W_j^TQ_{j+1} = 0$$

Eq (54) implies that T_j is the orthogonal projection of H onto the subspace spanned by the columns of W_j. Also if (θ, s) is an eigenpair of T_j, i.e. $T_js=s\theta$, then (λ, W_js) is an approximate eigenpair of H. A discussion on the accuracy of the approximation will be delayed until the spectral transformation Block Lanczos Algorithm is considered. Basically the Lanczos algorithm replaces a large and difficult eigenvalue problem involving H by a small and easy eigenvalue problem involving the block tridiagonal matrix T_j.

3.3 Spectral Transformation Block Lanczos Algorithm

Since our primary consideration is vibration problems, consider the eigenprobem posed by Eq (47) i.e.

$$M(K-\sigma M)^{-1}Mx = uMx$$

The Lanczos recurrence with block size p for solving Eq (47) is given by

Initialization

set $Q_o = 0$

set $B_1 = 0$

choose R_1 and orthonormalize the columns of R_1 to obtain Q_1 with $Q_1^T M Q_1 = I_p$

Lanczos Loop

For $j = 1, 2, 3, \ldots$

set $U_j = (K\text{-}\sigma M)^{-1}\,(MQ) - Q_{j-1}\, B_j^T$

set $A_j = U_j^T\,(MQ_j)$

set $R_{j+1} = U_j - Q_j A_j$

Compute Q_{j+1} and (MQ_{j+1}) such that

a) $Q_{j+1} B_{j+1} = R_{j+1}$

b) $Q_{j+1}^T\,(MQ_{j+1}) = I_p$

End Loop

Note that the algorithm as written requires only one multiplication by M per step and no factorization of M is required. The matrices Q_j are now M orthogonal, rather than orthogonal, i.e.

$$Q_j^T M Q_j = I \tag{55}$$

Also the Lanczos vectors are M orthogonal, i.e.

$$W_j^T M W_j = I$$

The recurrence formula in the Lanczos loop can also be written as

$$Q_{j+1} B_{j+1} = (K - \sigma M)^{-1} M Q_j - Q_j A_j - Q_{j-1} B_j^T \tag{56}$$

Now, as before, combining all j formulas of Eq (56) into one equation yields

$$(K - \sigma M)^{-1} M W_j = W_j T_j + Q_{j+1} B_{j+1} E_j^T \tag{57}$$

where W_j, T_j, and E_j are as defined in Eq (53). Premulitplying Eq (57) by $W_j^T M$ implies

$$W_j^T M\,(K - \sigma M)^{-1} M W_j = W_j^T M W_j T_j + W_j^T Q_{j+1} B_{j+1} E_j^T$$

i.e.

$$W_j^T M\,(K - \sigma M)^{-1} M W_j = T_j \tag{58}$$

since

$$W_j^T M W_j = I \quad \text{and} \quad W_j^T Q_{j+1} = 0$$

Eq (58) implies that T_j is the M-orthogonal projection of $(K - \sigma M)^{-1}$ onto the subspace spanned by the columns of W_j. The eigenvalues of T_j will approximate the eigenvalues of Eq (47). If (θ, s) is an eigenpair of T_j, then $(\theta, W_j s)$ will be an approximate eigenpair of Eq (47).

Recall that our main interest is in solving Eq (8). From Eq (48)

$$\theta = \frac{1}{\nu - \sigma}$$

$$\text{or} \quad \nu = \sigma + \frac{1}{\theta} \tag{59}$$

i.e. if θ is an approximate eigenvalue of T_j, then from Eq (59) ν is an approximate eigenvalue of Eq (8). Recall that the spectral transformation does not change the eigenvectors, therefore $y = W_j s$ is an approximate eigenvector for Eq (8).

Let's examine the approximations obtained by solving the block tridiagonal eigenvalue problem involving the matrix T_j. Let (θ, s) be an eigenpair of T_j i.e.

$$T_j s = s\theta$$

and let $y = \overline{W}_j s$. Then Premulitplying Eq (57) by M and post multiplying by s gives

$$\begin{aligned} M(K - \sigma M)^{-1} M W_j s - M W_j T_j s &= M Q_{j+1} B_{j+1} E_j^T s \\ M(K - \sigma M)^{-1} M y - M W_j s\theta &= M Q_{j+1} B_{j+1} E_j^T s \\ M(K - \sigma M)^{-1} M y - M y\theta &= M Q_{j+1} B_{j+1} E_j^T s \end{aligned} \tag{60}$$

Recall for any vector q, $\|q\|_{M^{-1}} = q^T M^{-1} q$ (Ref 4).

Therefore, taking the norm of Eq (60) and using Eq (55)

$$\begin{aligned} \|M(K - \sigma M)^{-1} M y - M y\theta\|_{M^{-1}} &= \|M Q_{j+1} B_{j+1} E_j^T s\|_{M^{-1}} \\ &= \|B_{j+1} E_j^T s\|_2 \equiv \beta_j \end{aligned} \tag{61}$$

Note that β_j is easily computed for each eigenvector s. It is just the norm of the p vector obtained by multiplying the upper triangular matrix B_{j+1} with the last p components of s.

From Ref 5 the error in eigenvalue approximations for the generalized eigenproblem is given by

$$\left|\frac{1}{\lambda-\sigma}-\theta\right| \le \frac{\|M(K-\sigma M)^{-1}My - My\theta\|_{M^{-1}}}{\|My\|_{M^{-1}}} = \beta_j \tag{62}$$

Thus β_j is a bound on how well an eigenvalue of T_j approximates an eigenvalue of Eq (47). Recall that if θ is an approximate eigenvalue of T_j, then from Eq (48)

$$\nu = \sigma + \frac{1}{\theta}$$

is an approximate eigenvalue of Eq (8). Consider

$$\begin{aligned}|\lambda-\nu| &= \left|\lambda-\sigma-\frac{1}{\theta}\right| \\ &= \frac{1}{\theta}\left|(\lambda-\sigma)\left(\frac{1}{\lambda-\sigma}-\theta\right)\right| \\ &\le \frac{1}{|\theta|}|\lambda-\sigma|\beta_j \le \frac{\beta_j}{\theta^2}\end{aligned} \tag{63}$$

Therefore $|\lambda-\nu| \le \frac{\beta_j}{\theta^2}$. Thus $\frac{\beta_j}{\theta^2}$ is a bound on how well the eigenvalues of Eq (47) approximate the eigenvalues of Eq (8).

3.4 An Analysis of the Block Tridiagonal Matrix T_j

The eigenproblem for T_j is solved by reducing T_j to a tridiagonal form and then applying the tridiagonal Q_L algorithm. The eigenextraction is accomplished in three steps:

1 An orthogonal matrix Q_T is found so that T_j is reduced to a tridiagonal matrix H, i.e.

$$Q_T^T T_j Q_T = H \tag{64}$$

2. An orthogonal matrix Q_H is found so that H is reduced to a diagonal matrix of eigenvalues, Λ, i.e.

$$Q_H^T H Q_H = \Lambda \tag{65}$$

3. Combining Eqs (64) and (65) gives

$$(Q_H^T Q_T)^T T_j (Q_T Q_H) = \Lambda \tag{66}$$

where $Q_T Q_H$ is the eigenvector matrix for T_j. The orthogonal matrices Q_H and Q_T are a product of simplex orthogonal matrices, Givens' rotations, $Q_{H_1} Q_{H_2} \cdots Q_{H_s}$ or $Q_{T_1} Q_{T_2} \cdots Q_{T_r}$. The algorithms used for steps (1) and (2) are standard and numerically stable algorithms drawn from the EISPACK collection of eigenvalue routines.

Note from Eq (61) that only the bottom p entries of the eigenvectors of T_j are needed for the evaluation of the residual bound. Therefore it is unnecessary to compute and store the whole eigenvector matrix for T_j. Only the last p components of the eigenvector matrix are computed.

The error bounds on the eigenvalues Eq (62) and (63) are used to determine which eigenvectors are accurate enough to be computed. At the conclusion of the Lanczos run the EISPACK subroutines are used to obtain the full eigenvectors of T_j. Then the eigenvectors for Eq (47) are found through the transformation

$$y = W_j s$$

3.5 Other Considerations in Implementating the Lanczos Algorithm.

The use of the block Lanczos algorithm in the context of the spectral transformation necessitates careful attention to a series of details:

a. The implications of M-orthogonality of the blocks

b. Block generalization of single vector orthogonalization schemes

c. The effect of the spectral transformation on orthogonality loss

d. The interactions between the Lanczos algorithm and the shifting strategy.

All of these issues are addressed in detail in Refs. 5,6.

3.6 The Block Lanczos algorithm as described in the previous sections was developed as a general purpose eigensolver for MSC NASTRAN (Ref 7). Boeing designed the software such that the eigensolver was independent of the form of the sparse matrix operations required to represent the matrices involved and their spectral transformations. The key operations needed were matrix-block products, triangular block solves and sparse factorizations. These, and the data structures representing the matrices, are isolated from the eigensolver. Therefore, the eigensolver code could be incorporated in different environments.

For this paper we tested the block Lanczos algorithm as incorporated in MSC NASTRAN and as further developed by Boeing and incorporated into code by Cray Research, Inc. The block Lanczos algorithm in MSC uses the factorization and solve modules which are standard operations in MSC. The Cray Lanczos code uses the Boeing eigensolver with matrix factorization, triangular solves, and matrix-vector products from the mathematical libraries supplied by Boeing computer services (BSCLIB-EXT). For vibration problems the CRAY code can be used with the stiffness and mass matrices, K and M, as generated by NASTRAN. NASTRAN is run to generate binary files containing the K and M matrices. These files are input files to the Cray code which calculates eigenvalues, checks the orthogonality of the eigenvectors, x, via $x'Kx$, calculates the Rayleigh quotient $x'Kx/x'Mx$ to compare with the computed eigenvalues, and calculates the norm of the eigenvector residual. In addition binary eigenvalue and eigenvector files output from the CRAY are suitable for input to NASTRAN for further processing if desired. Since the commercial (MSC) and the government COSMIC) NASTRANS do not give M and K in the same formats, they need to be reformatted before calling the CRAY code. CSAR-NASTRAN was used to represent NASTRAN on the CRAY XMP.

4.0 Test Problems

In this section several test problems were solved using the inverse power and FEER eigenvalue extraction methods in COSMIC NASTRAN, the Lanczos algorithm in MSC NASTRAN and the Lanczos algorithm as implemented by CRAY Research. These problems were chosen based on the complexity of the finite element model in terms of the kinds of elements used and the number of degrees of freedom. All methods as expected gave approximately the same numerical results. The only criterion used to compare the different methods was the number of seconds needed to reach a solution given that all problems were solved on the same platform, a CRAY XMP.

4.1 Problem 1 Square Plate

A square 200 in x 200 in plate in the x-y plane was modeled with QUAD4 elements only. Five meshes were defined. Details are given in Table 1. All elements were 1.0 in thick. Material properties were constant for all meshes. Each plate was completely fixed along the x-axis and the y-axis at x=200 in.

	MESH				
	10 x 10	20 x 20	30 x 30	40 x 40	50 x 50
Number of Grid Points	121	441	961	1681	2601
Number of Elements	100	400	900	1600	2500
Number-of Degrees of Freedom	515	2015	4515	8015	12515

Table 1: DETAILS OF THE FIVE MESHES DEFINED ON THE SQUARE PLATE

For all cases 5 frequencies were requested in the interval [0, 20hz]. Table 2 gives the results for the 10 x 10 plate, and Table 3 gives the results for the 50 x 50 plate. As expected within each case the numerical results from the different eigenextraction techniques are approximately the same. The differences in numerical results between the 10 x 10 case and the 50 x 50 case reflect the fineness of the mesh for the 50 x 50 case. Both Lanczos algorithms were run with a fixed block size of $p = 7$.

FREQUENCIES IN Hz

	1	2	3	4	5
COSMIC Inverse Power	6.2980	7.1720	11.6374	17.4440	18.3096
COSMIC FEER	6.2980	7.1720	11.6374	17.4440	18.3096
MSC Lanczos	6.2730	7.2173	11.7181	17.2125	18.3392
CRAY Lanczos	6.2730	7.2173	11.7181	17.2125	18.3392

Table 2: 10 x 10 SQUARE PLATE

FREQUENCIES IN Hz

	1	2	3	4	5
COSMIC Inverse Power	6.4048	7.6103	12.5487	17.6764	19.3642
COSMIC FEER	6.4048	7.6103	12.5487	17.6764	19.3642
MSC Lanczos	6.4054	7.6159	12.5599	17.6745	19.3739
CRAY Lanczos	6.4054	7.6159	12.5599	17.6745	19.3739

Table 3: 50 x 50 SQUARE PLATE

Table 4 gives the CPU time in seconds from the CRAY XMP needed to extract five frequencies for each case. Recall that the CRAY Lanczos algorithm needs to obtain the mass and stiffness matrices in binary form from NASTRAN. Thus the time given for this algorithm is the total time from two computer runs, i.e. the time to obtain the mass and stiffness matrices plus the time to run the Lanczos algorithm separately.

	MESH SIZE				
	10 x 10	20 x 20	30 x 30	40 x 40	50 x 50
COSMIC Inverse Power	14.734	50.936	97.801	197.769	328.830
COSMIC FEER	8.085	19.363	39.877	77.994	132.179
MSC Lanczos	4.783	13.641	30.973	59.283	103.188
CRAY Lanczos	4.174	11.170	23.785	45.433	78.009

Table 4: CPU TIME IN SECONDS TO OBTAIN 5 FREQUENCIES

Figure 2 is a plot of the degrees of freedom versus the CPU time in seconds on the CRAY for the four eigenvalue extraction techniques.

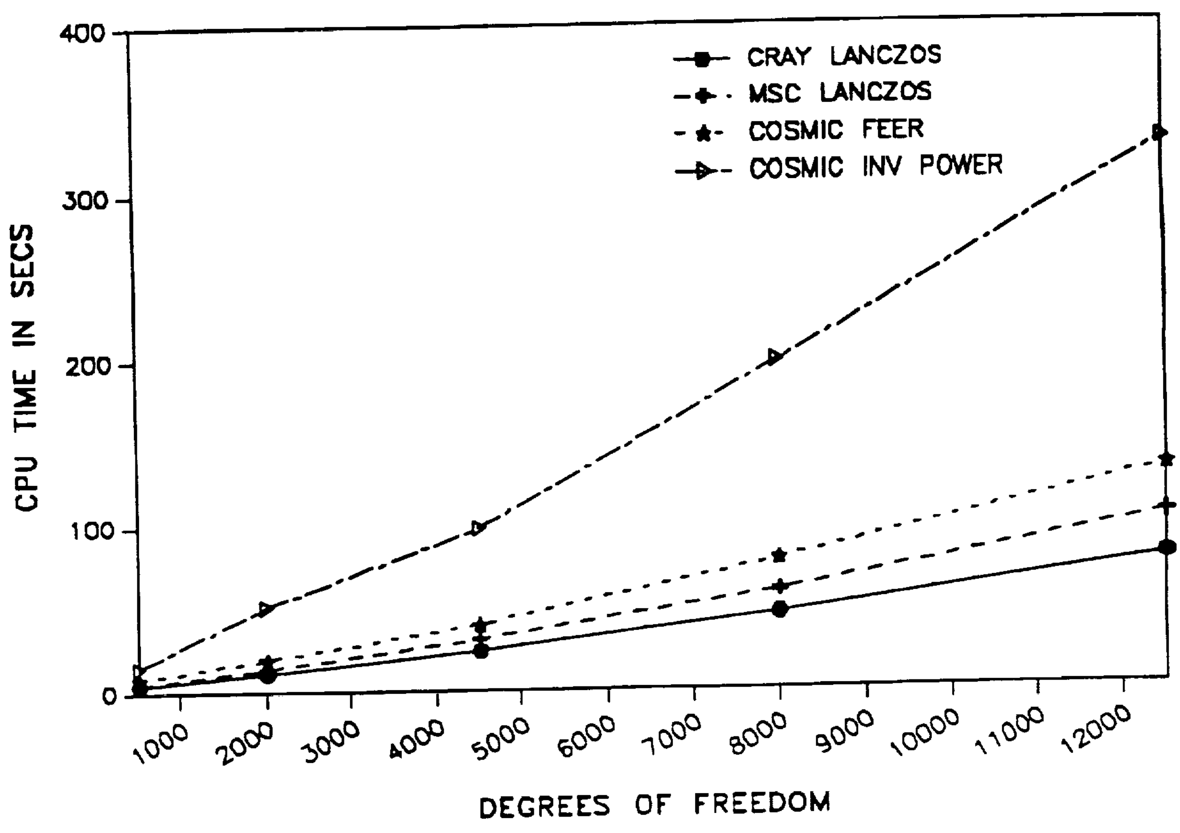

Figure 2: Degrees of Freedom versus CPU Time in Seconds.

4.2 Problem 2 Intermediate Complexity Wing

A three spar wing shown in Figure 3 was modeled with 88 grids and 158 elements of the following types: 62 QUAD4, 55 SHEAR, 39 ROD and 2 TRIA3. All elements varied in thickness or cross-sectional area. Material properties were the same for all elements. The wing was completely fixed at the root which left 390 degrees of freedom. Five frequencies were requested in the interval [0, 300hz]. Table 5 gives the frequencies calculated and the CPU time in seconds for the four eigenextraction algorithms. As for Problem 1 both Lanczos algorithms were run with a fixed block size of $p = 7$.

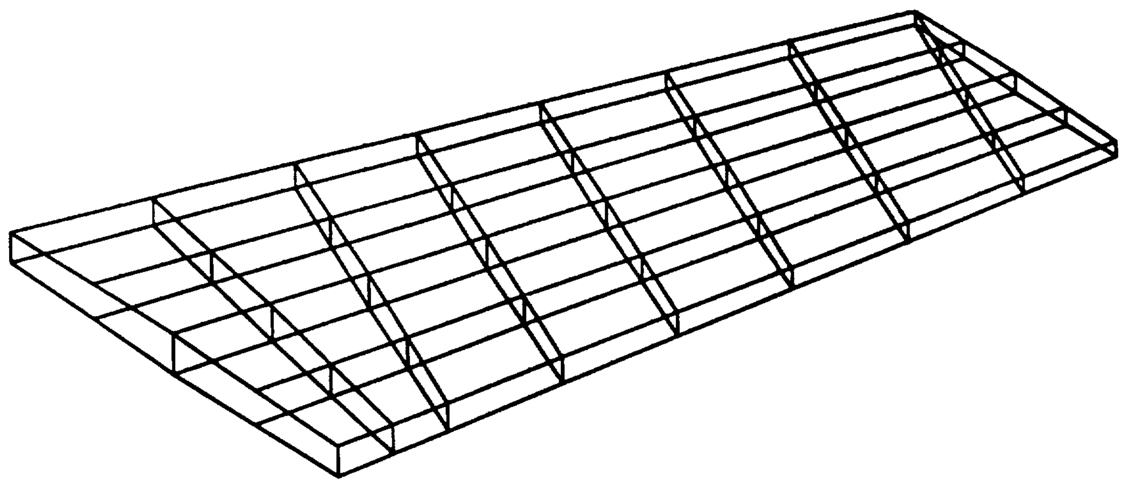

Figure 3: Intermediate Complexity Wing

	FREQUENCIES IN Hz					CPU TIME IN SECONDS
	1	2	3	4	5	
COSMIC Inverse Power	46.574	135.924	176.813	205.030	254.713	10.314
COSMIC FEER	46.574	135.924	176.813	205.030	254.713	8.085
MSC Lanczos	46.573	135.918	176.811	205.029	254.690	4.886
CRAY Lanczos	46.573	135.918	176.811	205.029	254.690	4.873

Table 5: INTERMEDIATE COMPLEXITY WING RESULTS

4.3 Problem 3 Radome

A composite radome shown in Figure 4 was modeled with 346 grids and 630 elements of the following types: 54 TRIA2, 284 BAR and 292 QUAD4. The QUAD4's were both isotropic and composite with 46 elements isotropic and 246 elements modeled as four cross-ply unsymmetric laminates of 40, 38, 36, and 32 layers, respectively. The radome was completely fixed at the base which left 1782 active degrees of freedom. Ten frequencies were requested in the interval [0,100 hz]. Table 7 gives the frequencies calculated and the CPU time in seconds for the four eigenextraction algorithms. Both Lanczos algoirthms were run with a fixed blocksize of $p = 7$.

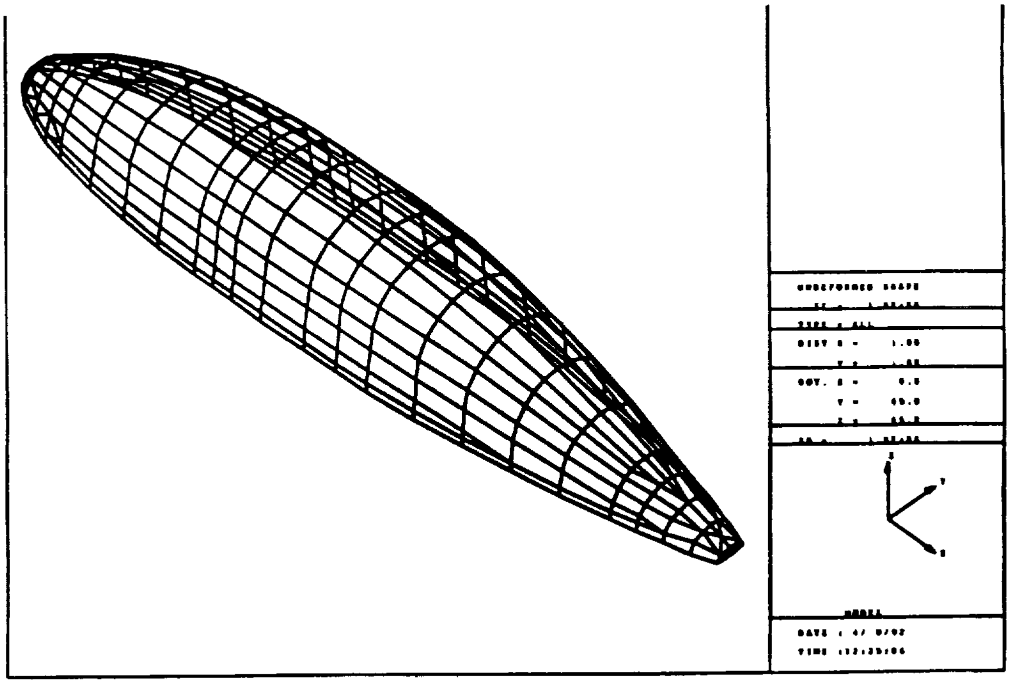

Figure 4: Radome

	FREQUENCIES IN Hz										CPU TIME IN SECS
	1	2	3	4	5	6	7	8	9	10	
COSMIC Inverse Power	56.325	67.946	69.290	81.486	90.835	90.971	92.074	92.410	93.365	101.441	63.986
COSMIC FEER	56.325	67.946	69.290	81.486	90.835	90.971	92.074	92.410	93.365	101.441	21.318
MSC Lanczos	56.068	66.958	68.213	80.843	89.715	90.248	90.768	91.676	92.365	98.729	17.768
CRAY Lanczos	56.068	66.958	68.213	80.843	89.715	90.248	90.768	91.676	92.365	98.729	13.854

Table 6: Radome Results

4.4 Problem 4 Satellite

A satellite shown in Figure 5 was modeled with 2295 grids and 1900 elements distributed as shown in Table 7.

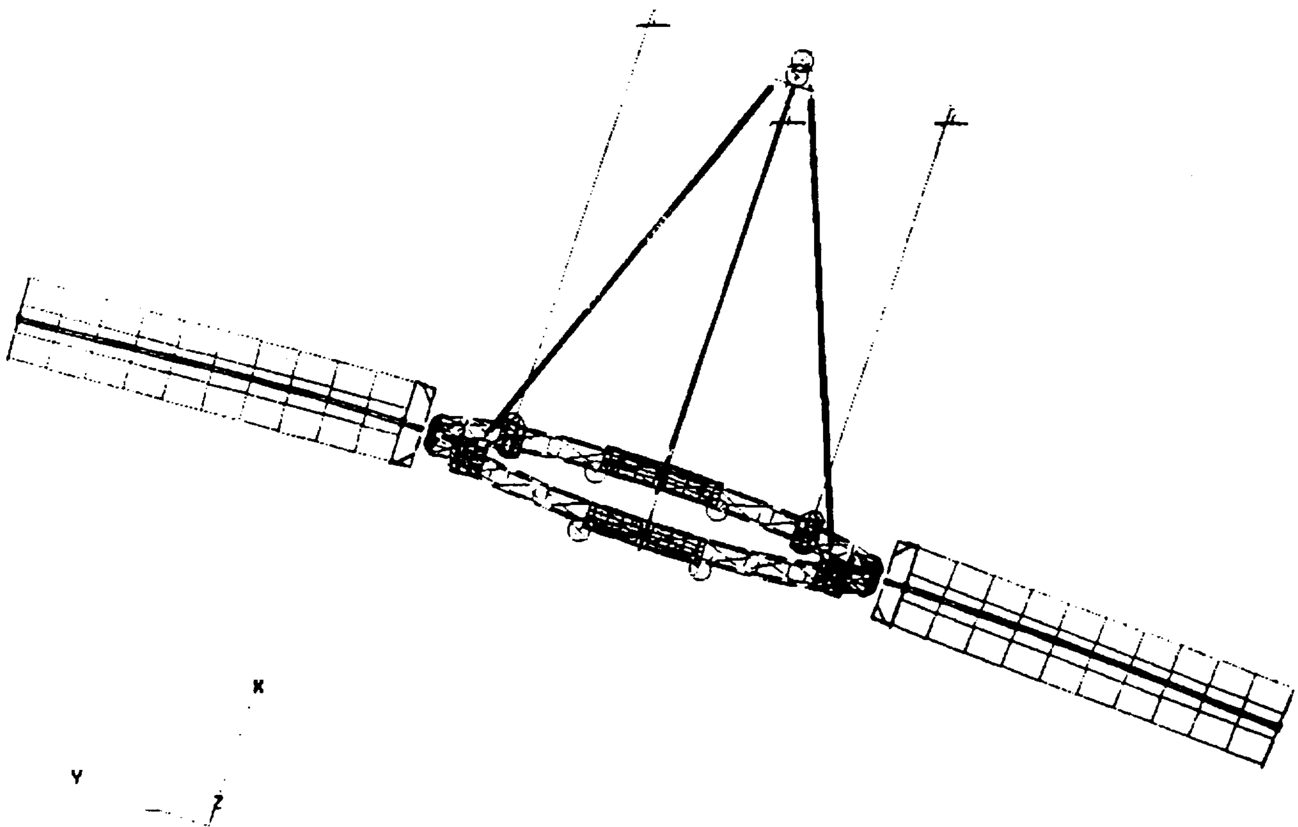

Figure 5: Satellite

ELEMENT TYPE

	ROD	BEAM	ELAS1	ELAS2	TRIA3	QUAD4	BAR	HEXA	PENTA	RBE2
Number of Elements	15	134	30	8	45	777	297	40	56	498

Table 7: Satellite Element Distribution

Sixteen different materials were referenced, and 34 coordinate systems were used. All elements varied in thickness and cross-sectional area, and concentrated masses were added to selected grids. The satellite has 5422 active degrees of freedom. Fifty frequencies were requested in the interval [0, 20hz]. Table 8 gives every fifth frequency calculated and the CPU time in seconds for the four eigenextraction algorithms. Again both Lanczos algorithms were run with a fixed block size of $p = 7$.

	FREQUENCIES IN Hz											CPU TIME IN SEC
	1	5	10	15	20	25	30	35	40	45	50	
COSMIC Inverse Power					NO SOLUTION IN 2000 SECS							
COSMIC FEER	.072	.313	1.497	1.663	2.419	5.414	9.000	10.974	13.328	17.474	19.758	294.759
MSC Lanczos	.072	.313	1.497	1.634	2.406	5.417	9.056	10.975	13.267	17.104	19.649	121.065
CRAY Lanczos	.072	.313	1.497	1.635	2.406	5.418	9.056	10.975	13.268	17.111	19.650	81.016

Table 8: SATELLITE RESULTS

4.5 Problem 5 Forward Fuselage - FS 360.0 - 620.0

A section of a Forward Fuselage from FS 360.0 to 620.0 shown in Figure 6 was modeled with 1038 grids and 3047 elements distributed as shown in Table 9.

Eleven different materials were referenced. All elements varied in thickness or cross-sectional area. The fuselage was fixed in the 123 directions at FS 620.0. The model had 6045 active degrees of freedom . Sixty frequencies were requested in the interval [0, 20hz]. Table 10 gives every fifth frequency calculated plus the last one and the CPU time in seconds for the four eigen-extraction algorithms. Both Lanczos algorithms were run with a fixed block size of p = 7.

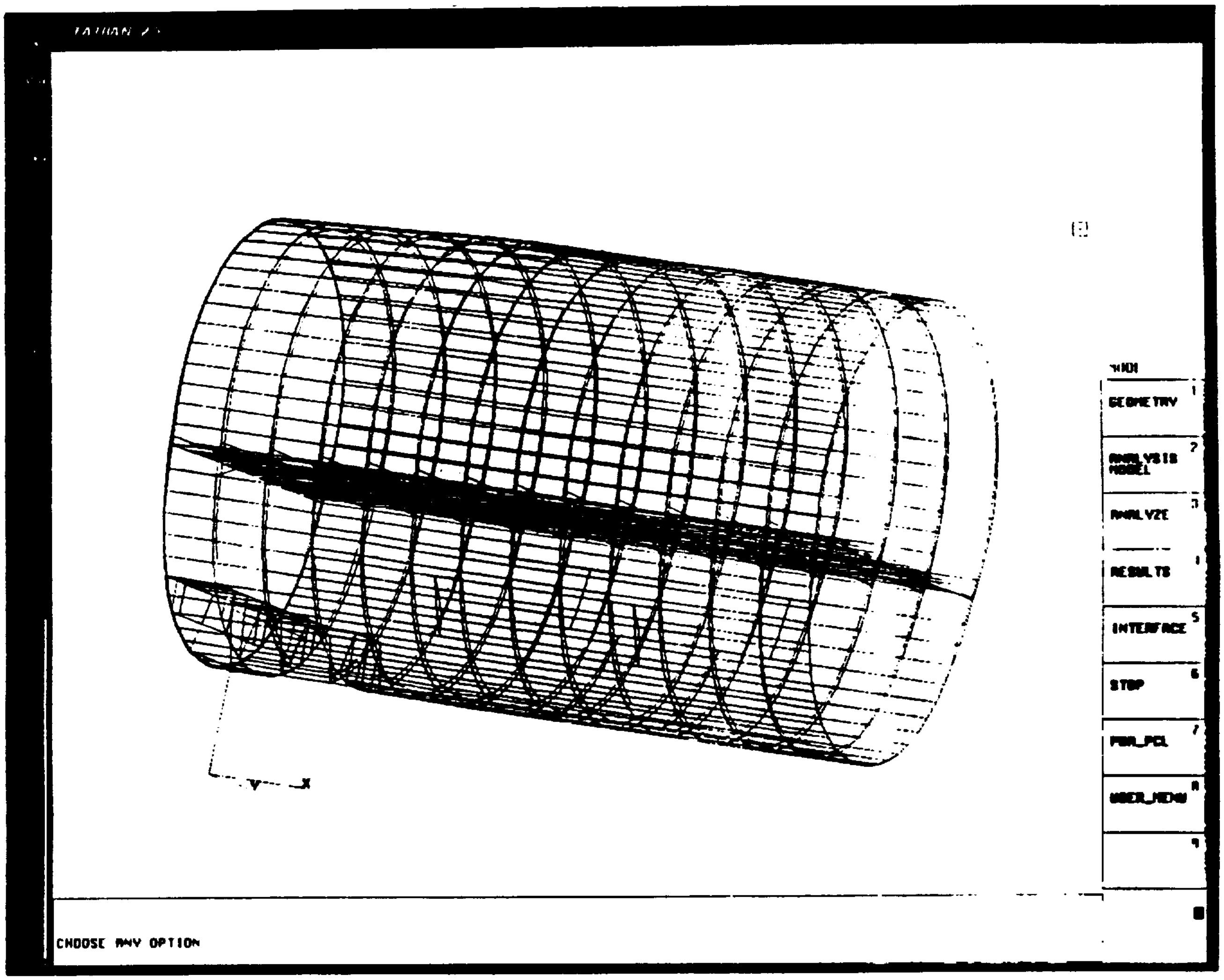

Figure 6: Forward Fuselage

ELEMENT TYPE

	BEAM	CONROD	SHEAR	TRIA3	QUAD4	BAR
Number of Elements	1141	885	395	15	572	39

Table 9: Forward Fuselage Element Distribution

FREQUENCIES IN Hz

	1	5	10	15	20	25	30	35	40	45	50	55	59	CPU TIME IN SECS
COSMIC Inv Power				NO SOLUTION IN 3000 SECS										
COSMIC FEER	.461	.819	2.093	3.090	5.577	7.467	12.247	15.175	16.097	17.515	18.183	19.403	22.658	180.348
MSC Lanczos	.462	.823	2.507	3.440	5.546	7.362	10.767	14.020	15.682	16.688	17.805	18.303	19.063	135.812
CRAY Lanczos	.462	.823	2.507	3.440	5.546	7.362	10.767	14.020	15.682	16.688	17.805	18.303	19.063	66.011

Table 10: Forward Fuselage Results

5.0 Summary and Recommendations

The current real eigenvalue analysis capability in NASTRAN in quite extensive and adequate for small and medium size problems. In particular the FEER Method's performance is reasonable at least for the problems tested in this paper. However, the Block Lanczos Method as implemented by BCS is more efficient for all the problems.

An analysis of Section 4 results clearly shows that the Block Lanczos Algorithm merits consideration for possible implementation into NASTRAN. Comparing CPU secs Table 4 implies that the CRAY Lanczos method runs 94% to 64% faster than the FEER method. Similarly from Tables 5, 6, 8 and 10 the CRAY Lanczos runs 66%, 54%, 260% and 177%, respectively, faster than the FEER method.

The comparisons are not near as striking when we consider the CRAY Lanczos and the MSC Lanczos. Comparing CPU seconds the CRAY Lanczos runs from .2% faster in Table 5 to 105.7% faster in Table 10. The difference in CPU time reported for these two methods can be attributed to two factors: (1) algorithm enhancements and (2) the Boeing Extended Mathematical Subprogram Library (BCSLIB-EXT) versus the standard mathematical modules in MSC. The CRAY Lanczos is based on [Ref 5] which is, most recent, dated July 1991. The MSC Lanczos is based on [Ref 6] which is dated 1986 plus subsequent updates by MSC. All problems were run under MSC NASTRAN Version 66a. Recent communications with Roger G. Grimes at Boeing, one of the developers of the shifted Block Lanczos algorithm, reveals that the Lanczos algorithm is continuously being refined and improved.

The problems chosen to test the four eigenextraction methods while diverse in terms of the number of degrees of freedom and element distribution were stable with no clusters of multiple eigenvalues. The multiple eigenvalue problem and its relation to the user chosen blocksize, *p*, is discussed in detail in [Ref 5]. The authors conclude that based on timing results for the selected problems, the shifted Block Lanczos Algorithm should be considered for possible implementation into NASTRAN.

Boeing Computer Services is reluctant to sell or lease their Block Lanczos routine to public domain programs such as COSMIC-NASTRAN or ASTROS. In view of this the authors recommend the following alternatives:

- Modify the FEER Method from a single vector Lanczos algorithm to a Block Lanczos algorithm.
- Obtain the Block Lanczos algorithm from an alternate source.
- Provide links for calling subroutines from the commercial math libraries such as the BCS or CRAY library.

REFERENCES

1. The NASTRAN Theoretical Manual, NASA SP-221, January 1981

2. Cullum, J. and Willoughby, R. A., "Computing Eigenvalues of Very Large Symmetric Matrices - An Implementation of a Lanczos Algorithm with No Reorthogonalization," Journal of Computational Physics, Vol. 44, No. 23, December 1991.

3. Cullum, J. and Willoughby, R. A., "A Survey of Lanczos Procedures for Very Large Real 'Symmetric' Eigenvalue Problems," Journal of Computational and Applied Mathematics, 12 & 13, 1985.

4. B. N. Parlett, "The Symmetric Eigenvalue Problem", Prentice-Hall Series in Computational Mathematics, Prentice-Hall Inc., Englewood Cliffs, N. J., 1980.

5. Grimes, R. G., Lewis, J. G., and Simon H., "A shifted Block LANCZOS Algorithm for Solving Sparse Symmetric Generalized Eigenproblems," Boeing Computer Services, AMS-TR-166, July 1991.

6. Grimes, R. G., Lewis, J.G., and Simon H., "The Implementation of a Block Shifted and Inverted Lanczos Algorithm for Eigenvalue Problems in Structural Engineering," Applied Mathematics Technical Report, Boeing Computer Services, ETA-TR-39, August 1986.

7. MSC/NASTRAN Version 67, Application Manual CRAY (UNICOS) Edition, The MacNeal-Schwendler Corporation, November 1991.