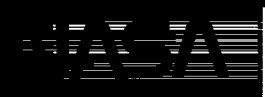
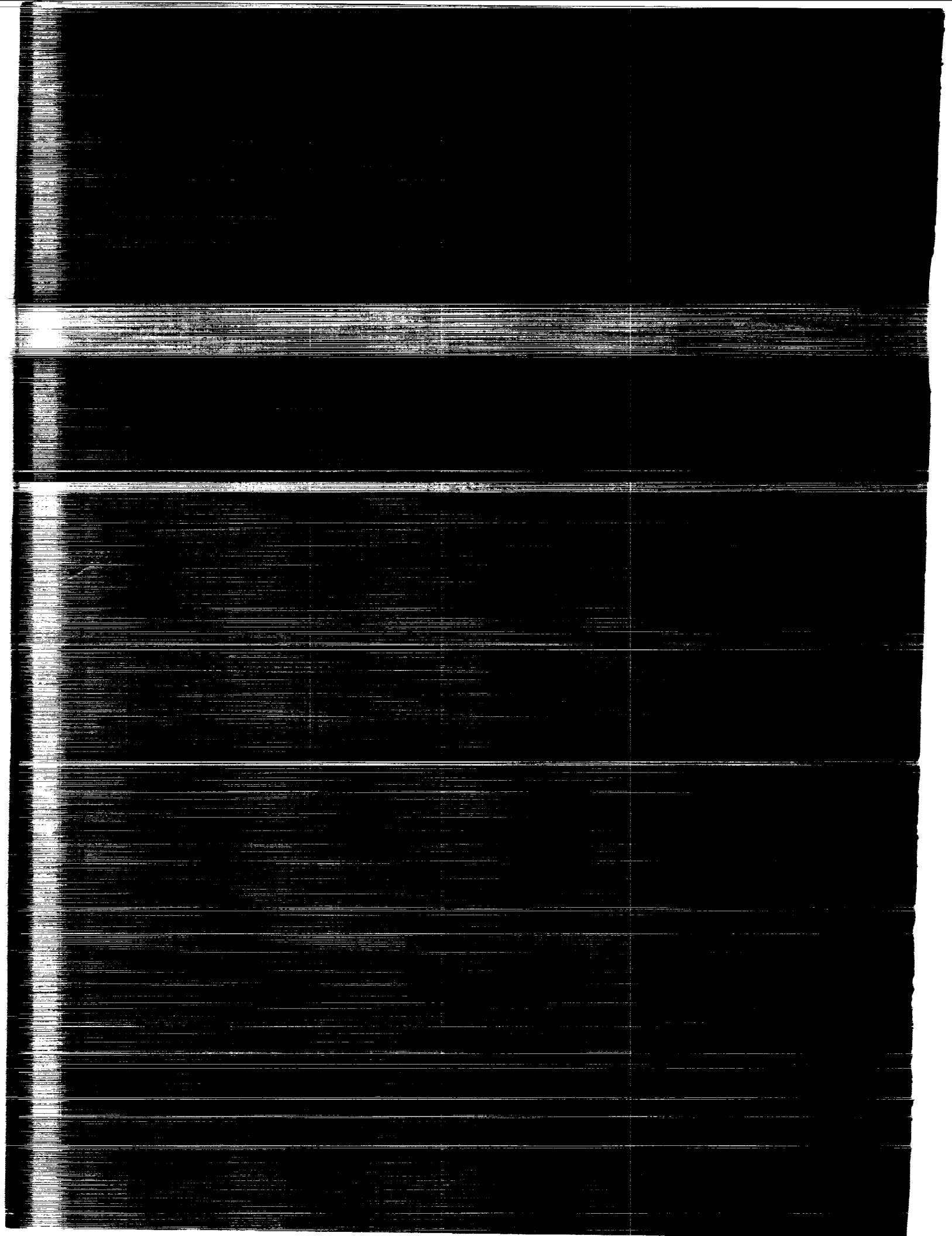
NASA Conference Publication 3203

Twenty-First NASTRAN® Users' Colloquium





ANALYTIC REVIEW FORM

(Initials)

Primary Record IPS # _____

___ Document should not receive analytic treatment

SUBSIDIARY ADD

Document page range ___ to ___ New subsidiary # ___
Document page range ___ to ___ New subsidiary # ___
Document page range ___ to ___ New subsidiary # ___

SUBSIDIARY DELETE/CORRECTION

Subsidiary #___
(IPS#_____)

___ Delete
Reason: ___ limited technical content
___ no separate authorship
___ context dependent

___ Adjust paging
___ Other _____

Subsidiary #___
(IPS#_____)

___ Delete
Reason: ___ limited technical content
___ no separate authorship
___ context dependent

___ Adjust paging
___ Other _____

Subsidiary #___
(IPS#_____)

___ Delete
Reason: ___ limited technical content
___ no separate authorship
___ context dependent

___ Adjust paging
___ Other _____

NASA Conference Publication 3203

Twenty-First NASTRAN® Users' Colloquium

Computer Software Management and Information Center
University of Georgia
Athens, Georgia

Proceedings of a colloquium held in
Tampa, Florida
April 26–30, 1993

NASA
National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Program
1993

FOREWORD

NASTRAN® (NASA STRUCTURAL ANALYSIS) is a large, comprehensive, nonproprietary, general purpose finite element computer code for structural analysis which was developed under NASA sponsorship and became available to the public in late 1970. It can be obtained through COSMIC® (Computer Software Management and Information Center), Athens, Georgia, and is widely used by NASA, other government agencies, and industry.

NASA currently provides continuing maintenance of NASTRAN through COSMIC. Because of the widespread interest in NASTRAN, and finite element methods in general, the Twenty-first NASTRAN Users' Colloquium was organized and held at the Sheraton Grand Hotel, Tampa, Florida on April 26 - April 30, 1993. (Papers from previous colloquia held in 1971, 1972, 1973, 1975, 1976, 1977, 1978, 1979, 1980, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991 and 1992 are published in NASA Technical Memorandums X-2378, X-2637, X-2893, X-3278, X-3428, and NASA Conference Publications 2018, 2062, 2131, 2151, 2249, 2284, 2328, 2373, 2419, 2481, 2505, 3029, 3069, 3111 and 3145.) The Twenty-first Colloquium provides some comprehensive general papers on the application of finite element methods in engineering, comparisons with other approaches, unique applications, pre- and post-processing or auxiliary programs, and new methods of analysis with NASTRAN.

Individuals actively engaged in the use of finite elements or NASTRAN were invited to prepare papers for presentation at the Colloquium. These papers are included in this volume. No editorial review was provided by NASA or COSMIC; however, detailed instructions were provided each author to achieve reasonably consistent paper format and content. The opinions and data presented are the sole responsibility of the authors and their respective organizations.

CONTENTS

IMPLEMENTATION OF MIXED FORMULATION ELEMENTS IN PC/NASTRAN

By

Harry G. Schaeffer

President
Thoroughbred CAE Software
Louisville, KY 40206

and

Chairman and Professor
Mechanical Engineering Department
University of Louisville
Louisville, KY 40292

SUMMARY

The purpose of this paper is to describe the implementation and use of a consistent family of two and three dimensional elements in NASTRAN. The elements which are based on a mixed formulation include a replacement of the original NASTRAN shear element and the addition of triangular quadrilateral shell elements and tetrahedral, pentahedral and hexahedral solid elements. These elements support all static loads including temperature gradient and pressure load. The mass matrix is also generated to support all dynamic rigid formats.

THEORETICAL CONSIDERATIONS

The principles of virtual and complementary virtual work allow us to formulate the elasticity problem in terms of either displacements or stresses. The formulation presented in (Ref. 1) provides us with the convenience of the displacement approach for statically indeterminate structures and the ease of stress recovery inherent in the stress approach. In the following we briefly outline the procedure for calculating the element stiffness matrix for the mixed formulation.

In order to derive the stiffness matrix we start with the complementary virtual work for the element which can be written as:

$$\delta W_c = \int_V \boldsymbol{\varepsilon}^T d\sigma dV - \int_{S_\sigma} \mathbf{v}^T \delta \mathbf{T} dS \qquad (1)$$

where T is the set of surface tractions on the boundary, S_σ.

The approach taken in the mixed formulation is to assume an equilibrium stress field σ within the element described in terms of a set of generalized parameters β; and to describe the boundary displacements **v** in terms of the grid point displacements **u**. The set of tractions **T** on the boundary are related to the stress components σ and the geometry of the element boundary so that it can be expressed in terms of the generalized coefficients β.

The equilibrium stresses are represented in the following form:

$$\sigma = \mathbf{Z}\beta \tag{2}$$

where the stress state does not include rigid body motion. The boundary traction can now be expressed in terms of the stress components and the unit normal to the boundary, which is only a function of geometry. It can thus be represented conceptually in the following form:

$$\mathbf{T} = \mathbf{L}\beta \tag{3}$$

Finally, the displacements along the boundary can be represented in terms of the grid point displacements as:

$$\mathbf{v} = \mathbf{N}\mathbf{u} \tag{4}$$

where **N** is a set of assumed shape functions that are appropriate for the order of the polynomial functions **Z** chosen to represent the equilibrium stresses.

Using the relationships for σ, **T**, and **v** and Hooke's law to relate σ and ε we can now write (1) as:

$$\delta W_c = \beta^T \mathbf{H}\,\delta\beta - \mathbf{u}^T \mathbf{R}\,\delta\beta = 0 \tag{5}$$

where:

$$\mathbf{H} = \int \mathbf{Z}^T \mathbf{E}^{-1} \mathbf{Z}\, dV \tag{6}$$

$$\mathbf{R} = \int \mathbf{N}^T \mathbf{L}\, dS \tag{7}$$

Since $\delta\beta$ is arbitrary it follows that:

$$\beta^T \mathbf{H} - \mathbf{u}^T \mathbf{R} = 0 \tag{8}$$

Solving for β gives:

$$\boldsymbol{\beta} = \mathbf{H}^{-1}\mathbf{R}^{T}\mathbf{u} \tag{9}$$

We can now write the internal strain energy in terms of displacements from which it can be seen that the stiffness matrix k is:

$$\mathbf{k} = \mathbf{R}^{T}\mathbf{H}^{-1}\mathbf{R} \tag{10}$$

In the next section the set of equilibrium stresses assumed for each of the elements that is included in the PC/NASTRAN element library is described.

Assumed Stress Fields

The assumed stress field used for the three dimensional stress field elements is:

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{Bmatrix} = \begin{bmatrix} \beta_1 & 0 & \beta_7 & \beta_8 & 0 & \beta_{16} & 0 \\ \beta_2 & \beta_9 & 0 & \beta_{10} & 0 & 0 & \beta_{17} \\ \beta_3 & \beta_{11} & \beta_{12} & 0 & \beta_{18} & 0 & 0 \\ \beta_4 & 0 & 0 & \beta_{13} & 0 & 0 & 0 \\ \beta_5 & \beta_{14} & 0 & 0 & 0 & 0 & 0 \\ \beta_6 & 0 & \beta_{15} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ y \\ z \\ xy \\ yz \\ zx \end{Bmatrix} \tag{11}$$

where only the coefficients terms 1-6 are used for the constant stress tetrahedron which is called the TETRA element in PC/NASTRAN. Similarly the stress field for the two dimensional stress field membrane and bending force and moment resultants are:

$$\begin{Bmatrix} \mathbf{N}_x \\ \mathbf{N}_y \\ \mathbf{N}_{xy} \end{Bmatrix} = \begin{bmatrix} \beta_1 & 0 & \beta_4 \\ \beta_2 & \beta_5 & 0 \\ \beta_3 & 0 & 0 \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} \tag{12}$$

and

$$\begin{Bmatrix} \mathbf{M}_x \\ \mathbf{M}_y \\ \mathbf{M}_{xy} \\ \mathbf{Q}_x \\ \mathbf{Q}_y \end{Bmatrix} = \begin{bmatrix} \beta_1 & \beta_6 & \beta_4 \\ \beta_2 & \beta_5 & \beta_7 \\ \beta_3 & \beta_9 & \beta_8 \\ \beta_6 - \beta_8 & 0 & \beta_{10} \\ \beta_7 - \beta_9 & \beta_{11} & 0 \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} \tag{13}$$

respectively. All of the coefficients in equations (12) and (13) are used for the quadrilateral plate element which is called the QUAD4. However only the constant terms 1-3 in equation (12) and 1-7 in equation (13) are used for the triangular element which is called the TRIA3. Shell behavior is represented as the the sum of membrane and bending behavior for both elements.

IMPLEMENTATION

An early decision was made to replace the original two and three dimensional elements with a consistent family of elements rather than to add to the existing family. PC/NASTRAN thus no longer includes the TRIMEM, QDMEM, HEXA1, HEXA2 etc. These have been replaced with

TRIA3	A triangular shell element with three vertex grid points
QUAD4	A quadrilateral shell element with four vertex grid points
TETRA	A tetrahedral solid element with four vertex grid points
PENTA	A pentahedral solid element with six vertex grid points
HEXA	A hexahedral solid element with eight vertex grid points

Element Matrix Generation

The element subroutines for the generation of element stiffness, mass and stress matrices are called by EMGPRO in the EMG module. The stiffness and mass matrices together with their directory entries are written using EMGOUT for later use by the Element Matrix Assembler (EMA). In addition the stress matrices and their directory is written out for subsequent use in generating thermal loads and in recovering element stresses.

Element Load Generation

The calculation of element-dependent loads including thermal loading which is specified by the standard NASTRAN thermal load Bulk Data and the grid point forces due to pressure load requires access to the element stress matrix and element geometry, respectively. Existing routines were modified to include the new elements and a new capability for generating grid point forces from surface pressure data was implemented. The associative Bulk Data is called the PLOAD4 which allows the user to define a surface traction with respect to either element of the global set of coordinates.

Elements Stresses and Forces

The SDR2 module was modified to accept the stress matrix and directory files produced in EMG. The stress recovery subroutines were written to interface with subroutine SDR2E.

Output Routines

The OFP module was modified to print the element stresses and forces for the new elements. Since the stress output is easily calculated at any point in the domain of the element, the stress and element forces are printed at the element centroid and at each vertex point.

In addition to the standard Output File Processor, separate binary files for each behavioral variable selected in Case Control can be created as a user option. The data structure of each binary file closely follows that of the associated file that is created for the OFP. The benefit in having the binary files is they can be read directly in binary format rather than parsing the ASCII output print file as many post processor programs do, thereby leading to a great speed increase especially for large print files. Another benefit is a reduction in the computer disk storage resources required to store the output.

Other Modifications

Several additional modifications were made to PC/NASTRAN to improve the user friendliness and efficiency of the analysis program. These are:

1. Grid Point Resequencing

   Grid point resequencing is automatically executed as a default but may be bypassed at user option. The resequencing strategies available to the user include Reverse Cuthill-McKee and Gibbs-Poole-Stockmeyer.

2. Automatic Constraint Generation

   In order to remove unconnected degrees of freedom a procedure is introduced to determine whether a singularity at the grid point level exists in the assembled stiffness matrix. If one does exist the automatic constraint generator determines whether a single point constraint or multiple point constraint equation is required to remove the constraint. The USET is updated accordingly and if the constraint is an MPC the associated data are written to a file and added to any MPC constraints selected by Case Control and those defined by rigid elements.

   The automatic MPC capability means that grid point singularities

which do not align with displacement coordinate degrees of freedom are handled correctly. The improvement can be demonstrated easily using a single rod element whose axis is not aligned with the displacement coordinate system as described in (Ref. 2).

3. Modified Givens Procedure

As new users of NASTRAN can attest, Fatal Error 3053 – MAA is singular is rather esoteric to the uninitiated. For the initiates it means that Givens Method for eigenvalue extraction has been selected. The associated transformation of the eigenvalue problem to standard form requires that the mass matrix be non-singular.

It can be time consuming to determine the set of massless degrees of freedom which must be removed by static condensation prior to using Givens method. An alternative is to reformulate the eigenvalue problem using a shift point so that the matrix is to be decomposed is always nonsingular. This method is called Modified Givens.

Dynamic Solutions

The solution sequences for normal modes, transient dynamic response and frequency response have been modified as required for the new elements. The eigenvalue solution options have been verified by solving for the modes and frequencies of several test models. In general, the results for the eigenvalues are identical for Givens, Modified Givens and the inverse power methods. Testing also shows that Givens and modified Givens will handled approximately 250 degrees of freedom before spilling.

The transient response and frequency response algorithms for both the modal and direct formulations produce results that agree well with those obtained from other NASTRAN implementations. At this time the random response capability has not been implemented.

RESULTS AND CONCLUSIONS

The implementation of mixed formulation elements in PC/NASTRAN has shown that:

1. NASTRAN is a powerful test bed for the development of computational structural mechanics algorithms.

2. PC/NASTRAN provides a low-cost powerful computational environment on Personal Computers.

3. The mixed formulation elements generally equal the performance of displacement-based elements with the same number of vertex grid points.

REFERENCES

1. Lee, S. W. and Pian, T. H. H., '*Improvement of plate and shell finite element by fixed formulation*', AIAA Journal, 16,29-34(1978).

2. Schaeffer, H. G., *A Guide to Finite Element Analysis Using PC/NASTRAN*, Thoroughbred CAE Software, Louisville, KY 1992.

52-39
190572
p. 9

N94-17829

IMPROVED OMIT SET DISPLACEMENT RECOVERIES IN DYNAMICS ANALYSIS

Tom Allen
Greg Cook
Bill Walls

McDonnell Douglas Aerospace
Huntsville Division

ABSTRACT

Two related methods for improving the dependent (OMIT set) displacements after performing a Guyan reduction are presented. The theoretical bases for the methods are derived. The NASTRAN DMAP ALTERs used to implement the methods in a NASTRAN execution are described. Data are presented that verify the methods and the NASTRAN DMAP ALTERs.

1.0 INTRODUCTION

A NASTRAN user is faced with two major challenges when solving a dynamic eigenvalue problem. First, an eigenvalue solution is expensive to perform for most structural problems encountered in engineering analysis, and second, many more degrees of freedom (DOF) are required to define a structure's elastic properties than are required to define its inertial properties.

A popular method for meeting these challenges is to reduce the problem size using Guyan reduction (Reference 1). Guyan reduction allows the user to preserve the elastic properties of the problem set while reducing the problem size to one that is more manageable for a dynamic eigenvalue analysis. At the same time, the mass properties are also condensed with some penalty associated with the redistribution of mass from the coordinates eliminated during the Guyan reduction. The present paper describes two approaches that correct the inaccuracies caused by the condensation of the mass matrix without unduly affecting the solution time.

The theoretical development of the improvement methods is provided in Section 2. Section 3 describes the NASTRAN DMAP ALTERs used to implement the algorithms used for both methods. Verification of the two methods, the second of which is a refinement of the first, is presented in Section 4. Conclusions and recommendations are provided in Section 5.

2.0 THE IMPROVEMENT METHOD

We begin by deriving the Guyan reduction scheme.

The dynamic eigenvalue problem is given by the equation

$$([K] - \lambda[M])\{\phi\} = 0 \qquad (1)$$

where

K = the structural stiffness matrix
M = the structural mass matrix
λ = the system eigenvalue
ϕ = the eigenvector or modal displacements.

We can partition Equation 1 into independent DOF, designated in NASTRAN as the analysis set, or A-set, and dependent DOF, designated as the OMIT set, or O-set. After performing this operation Equation 1 becomes

$$\left(\begin{bmatrix} \bar{K}_{aa} & K_{ao} \\ K_{ao}^T & K_{oo} \end{bmatrix} - \lambda \begin{bmatrix} \bar{M}_{aa} & M_{ao} \\ M_{ao}^T & M_{oo} \end{bmatrix} \right) \begin{Bmatrix} \phi_a \\ \phi_o \end{Bmatrix} = 0 \tag{2}$$

where the subscript "a" denotes A-set DOF and the subscript "o" denotes O-set DOF.

Looking at the lower partition of Equation 2 we can say

$$K_{ao}^T \phi_a + K_{oo} \phi_o - \lambda M_{ao}^T \phi_a - \lambda M_{oo} \phi_o = 0 \tag{3}$$

The Guyan reduction method (Reference 1) makes the assumption that the inertial forces on the O-set displacements are much less important than the elastic forces transmitted by the A-set displacements. A constraint equation for Guyan reduction can be derived by ignoring the mass terms in Equation 3. The resulting constraint equation is given by

$$\phi_o = G_o \phi_a \tag{4}$$

where

$$G_o = -K_{oo}^{-1} K_{ao}^T \tag{5}$$

This relationship constitutes a Ritz transformation of the eigenvalue problem. The transformation written in terms of the full displacement set is

$$\{\phi\} = \begin{Bmatrix} \phi_a \\ \phi_o \end{Bmatrix} = [G]\{\phi_a\} = \begin{bmatrix} I \\ G_o \end{bmatrix} \{\phi_a\} \tag{6}$$

Using this Ritz transformation, the reduced mass and stiffness matrices become

$$[M_{aa}] = [G]^T [M][G] \tag{7}$$

and

$$[K_{aa}] = [G]^T [K][G] \tag{8}$$

The mass of the system is redistributed based upon the elastic connections between the O-set DOF and the A-set DOF as shown in Equation 7.

The reduced mass and stiffness matrices shown in Equations 7 and 8, are then used to compute the eigenvalues and the A-set displacements of the reduced system. Once the A-set displacements have been computed, the Guyan reduction transformation of Equation 4 is used to recover the O-set displacements. This back transformation ignores the inertial terms of the O-set displacements.

An improved back transformation for ϕ_o can be found using Equation 3 (see Reference 2). For mode i, this back transformation is given by

$$\{\phi_o\}_i = -[K_{oo} - \lambda_i M_{oo}]^{-1}[K_{ao}^T + \phi_o - \lambda_i M_{ao}^T]\{\phi_a\}_i \tag{9}$$

Though Equation 9 will yield improved results, the first term on the right hand side must be inverted for each mode calculated during the eigenvalue analysis, a computationally inefficient process. Clearly, a more direct substitution would make the O-set displacement recovery more efficient.

Recasting Equation 3 for all the computed modes, we get

$$K_{ao}^T\phi_a + K_{oo}\phi_o - M_{ao}^T\phi_a \underline{\lambda} - M_{oo}\phi_o \underline{\lambda} = 0 \tag{10}$$

where $\underline{\lambda}$ is a square matrix with the system eigenvalues along the diagonal. Solving for the ϕ_o displacements that are not multiplied by $\underline{\lambda}$, we get

$$G_o\phi_a + K_{oo}^{-1}M_{ao}^T\phi_a \underline{\lambda} + K_{oo}^{-1}M_{oo}\phi_o \underline{\lambda} = \phi_o \tag{11}$$

From Equation 11 we can see that a closed form solution for ϕ_o does not exist. It is possible, however, to use Equation 11 to obtain an improved approximation to ϕ_o.

A first approximation to ϕ_o can be determined by using the O-set displacements recovered by Equation 4, or

$$\phi_o^{(1)} = G_o\phi_a \tag{12}$$

Substituting these O-set displacements into Equation 11 yields

$$G_o\phi_a + K_{oo}^{-1}M_{ao}^T\phi_a \underline{\lambda} + K_{oo}^{-1}M_{oo}\phi_o^{(1)} \underline{\lambda} = \phi_o^{(2)} \tag{13}$$

where $\phi_o^{(2)}$ are the corrected O-set displacements. These corrected displacements can be substituted back into Equation 13 for $\phi_o^{(1)}$ and a better approximation, $\phi_o^{(3)}$, can be computed. This process can be repeated until the displacements at the $(i+1)$ iteration are the same as the displacements at the i^{th} iteration. These "super" improved displacements will be identical to those computed using Equation 9, and can be determined without the computational penalty associated with inverting an O-set by O-set sized matrix for each mode.

To summarize, three methods for recovering the O-set displacements after performing the Guyan reduction and the reduced eigenvalue analysis have been presented . These three methods are:

1) Standard Guyan reduction recovery using Equation 4, henceforth designated as Guyan displacements.
2) Improved O-set displacement recovery using Equations 12 and 13, henceforth designated as improved displacements.
3) Successively iterated improved O-set displacements using Equation 13, henceforth designated as "super improved" displacements.

The reader will note that the A-set displacements are identical for all three methods described above. It is assumed that the eigenvalues and the A-set displacements computed during the eigenvalue analysis are "accurate". In other words, the accuracy (or inaccuracy) of the Guyan reduction itself is not in question.

Thus far, we have discussed improvements only in the O-set displacements. More importantly, any quantity computed using these O-set displacements, such as element forces or element stresses, will also be improved by methods 2 and 3.

The theory and methodology for improving the O-set displacements has been provided. The following section describes the implementation of the improved displacement recoveries in NASTRAN.

3.0 IMPLEMENTATION IN NASTRAN

With the methodology in hand, the implementation in NASTRAN becomes an exercise in defining the data blocks and the NASTRAN DMAP modules required to perform the desired operations. The DMAP ALTER sequences used to recover the improved displacements are provided in Figure 1. The first ALTER places the UPARTN module following the SMP1 module while the second ALTER places the DMAP modules used to recover the improved displacements after the SDR1 module. The user controls the recovery method with the parameters defined in the DMAP ALTERs. The allowable parameter values and the resulting action taken are provided in Table 1. Note that if no A-set is defined, the O-set recovery section is skipped.

```
$
$ DMAP Alter to obtain required matrices for improvement. Place after the SMP2 Module.
ALTER ii  $ where ii = DMAP statement number of Module SMP2
UPARTN          USET,MFF/,MAOT,,MOO/*F*/*A*/*O*  $
$
$ DMAP Alter to perform O-set displacement improvement. Place after the SDR1 Module.
ALTER jj  $ where jj = DMAP statement number of Module SDR1
COND            SKIPIM,OMIT $
$
$ This PARAM defines whether Guyan recovery or improvement
$ recovery is to be performed (NOIMP < 0, Guyan recovery)
PARAM           //*NOP*/NOIMP = -1  $
COND            SKIPIM,NOIMP $
$
$ This PARAM defines what recovery improvement will be performed
$ If NREPT = 0, improve once, NREPT > 0, iterate NREPT times
PARAM           //*NOP*/NREPT =10  $
$
$ MATGEN creates a square matrix from the LAMA table
MATGEN          LAMA/MLAMA/3/2 $
MPYAD           GO,PHIA,/PHIO/0/1/0/  $
FBS             LOO,,MAOT/C1/1/1  $
SMPYAD          C1,PHIA,MLAMA,,,PHIO/A/3///1  $
FBS             LOO,,MOO/B/1/1  $
LABEL           IMPRV  $
SMPYAD          B,PHIO,MLAMA,,,/C/3///1  $
ADD             A,C/PHIO/(1.0,0.0)/(1.0,0.0)  $
REPT            IMPRV,NREPT  $
UMERGE          USET,PHIA,PHIO/PHIF/*F*/*A*/*O*  $
UMERGE          USET,PHIF,/PHIN/*N*/*F*/*S*  $
MPYAD           GM,PHIN,/PHIM/0/1/0/  $
UMERGE          USET,PHIN,PHIM/PHIG/*G*/*N*/*M*  $
LABEL           SKIPM  $
```

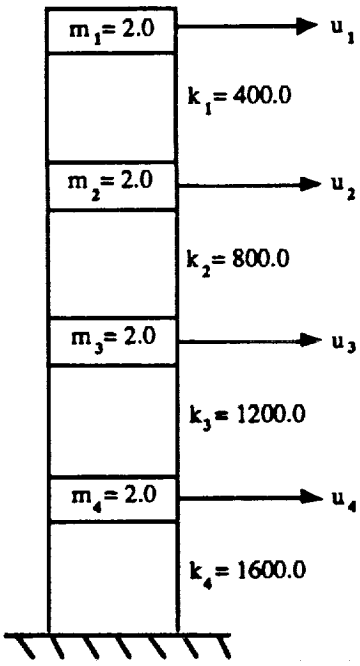
Figure 1. O-set Displacement Improvement DMAP ALTERs

Table 1. DMAP Parameter Settings		
Execution Type	NOIMP	NREPT
No A-set	N/A	N/A
Guyan	-1	N/A
Improved	0	0
Super Improved	0	# repetitions

Once the O-set displacements have been recovered, the rest of the standard solution sequence is executed. This allows the user to define all data recoveries using the familiar NASTRAN Case Control Deck commands. Displacements, element forces, element stresses, or any other user requested data will be printed and handled in the normal fashion. No special provisions are required to view the improved data.

4.0 METHOD VERIFICATION

Two sample problems were created to verify the method and the DMAP described in Section 3. The first sample problem consists of a simple four story building. This problem was used to verify the methodology and the DMAP ALTERs shown in Figure 1. The second problem consists of a 3600 DOF substructured model. Element forces for this model were recovered from a transient response analysis using the three O-set displacement recovery methods and compared to the benchmark element forces obtained when no Guyan reduction was performed. These sample problems verify the improvement methods and the DMAP ALTERs.



$$K = \begin{bmatrix} 400 & -400 & 0 & 0 \\ -400 & 1200 & -800 & 0 \\ 0 & -800 & 2000 & -1200 \\ 0 & 0 & -1200 & 2800 \end{bmatrix}$$

$$M = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$G_o = \begin{bmatrix} 1/3 & 2/3 \\ 0 & 3/7 \end{bmatrix}$$

$f_1 = 1.278$ Hz

A-set DOF = u_1 and u_3

Figure 2. Simplified Four Story Building

The four story building used for sample problem 1 is shown in Figure 2. This problem was selected because it is easily represented with NASTRAN elements and may be solved using the NASTRAN program. It may also easily be solved by hand so that the data produced by the DMAP ALTERs can be verified. Data were recovered for the first mode only.

Table 2 presents the O-set displacements for the three methods as well as the unreduced benchmark displacements. The data in Table 2 were recovered from NASTRAN using the DMAP ALTERs described in Section 3. The reader can easily verify that the Guyan results are identical to those recovered by hand using Equation 4, the improved results are identical to those recovered by hand using Equation 11, and the super improved data are identical to those recovered by hand using Equation 9. These data verify the DMAP sequence described in Section 3.

Table 2. Displacement Comparison

Disp.	Guyan	Improved	Super Improved*	Benchmark (No A-set)
u_1	1.0000	1.0000	1.0000	1.0000
u_2	0.6015	0.6681	0.6764	0.6775
u_3	0.4023	0.4023	0.4023	0.4069
u_4	0.1724	0.1806	0.1810	0.1828
MAC	0.99730	0.99995	0.99998	N/A

* These data were recovered using 10 iterations

The Modal Assurance Criterion (MAC) defined in Reference 4 is used to measure the accuracy of the eigenvectors provided in Table 2. MAC values will vary between zero, indicating no correlation between modes, to unity, indicating perfect correlation between modes. Based on the MAC values, it is clear that both improvement methods produce better O-set displacements than the standard Guyan recovery method produces alone.

The advantage of using the improved O-set recovery methods is clearer when element data, e.g. element forces or stresses, are compared. The modal spring forces for all three O-set displacement recovery methods are compared to the benchmark data in Table 3. From this it is clear that the improved displacements produce spring forces that are vastly superior to those of Guyan reduction.

Based on this simple problem, the displacements can be dramatically improved by using the methods described in Section 2. The next sample problem will show this more clearly.

The second sample problem uses the 3600 DOF Spacelab Pallet model shown in Figure 3. A transient response analysis was performed with this model in two configurations, an unreduced configuration and a Guyan reduced configuration. Transient element forces of all the bar elements were recovered using four distinct PHASE3 executions, i.e. no A-set, Guyan, improved, and super improved.

Table 3. Spring Force Comparison				
Spring Forces	Guyan	Improved	Super Improved	Benchmark (No A-set)
F_1	159	133	129	129
F_2	159	213	219	217
F_3	276	266	266	269
F_4	276	289	290	293
ΔF_1	30	4	0	N/A
ΔF_2	-58	-4	2	N/A
ΔF_3	7	-3	-3	N/A
ΔF_4	-17	4	-3	N/A

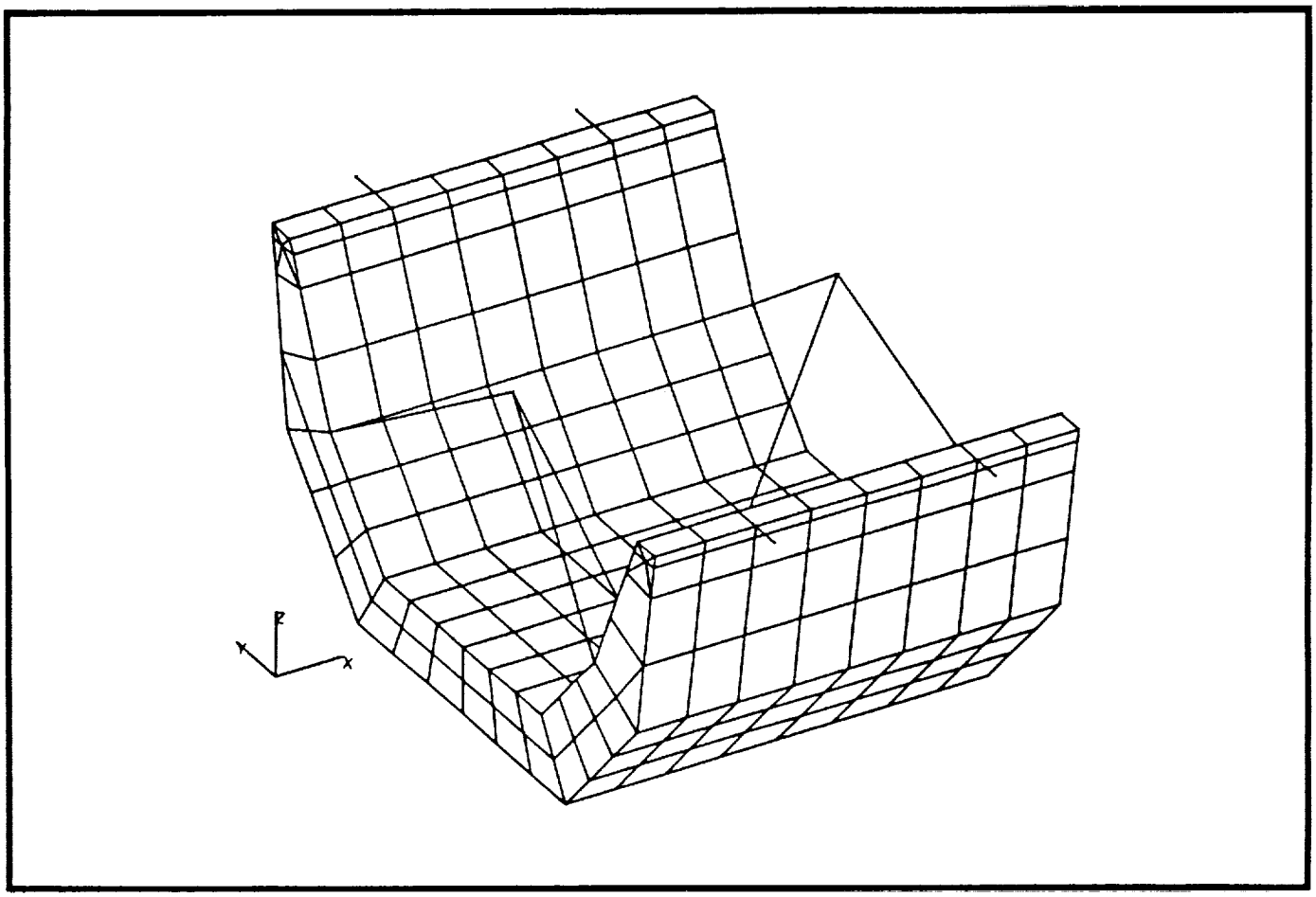


Figure 3. NASTRAN Model for Sample Problem 2

The maximum absolute values for all of the bar forces for the three recovery methods were compared to the benchmark case. These comparisons are shown in Table 4. The data are arranged according to a percentage difference range. For each of the recovery methods, the percentage of the forces falling within this range as well as the maximum difference between the benchmark data and the data produced by the three recovery methods within this delta percentage range are provided.

For example, in the range between two and five percent, 8.53 percent of the forces from the Guyan recovery method fell within this range with the maximum difference between the Guyan recovered data and the benchmark data being 397. For the improved recovery method, only 0.10 percent of the forces fell into this range with a maximum difference between the benchmark and the improved data being 5. The percentage of items falling in this range for the super improved method was 0.09, with a maximum delta of 7.

Table 4. Bar Element Force Comparisons for Sample Problem 2

Δ% Range	Guyan		Improved		Super Improved*	
	Percentage in Range	Maximum |Δ|	Percentage in Range	Maximum |Δ|	Percentage in Range	Maximum |Δ|
0-2	89.05	1045	99.90	102	99.76	114
2-5	8.53	397	0.10	5	0.09	7
5-10	1.48	48	0.00	0	0.03	4
10-25	0.60	82	0.00	0	0.00	0
25-50	0.03	0	0.00	0	0.00	0
>50	0.32	2281	0.00	0	0.13	36

* These data were recovered using 10 iterations

As was the case for the simplified model used for sample problem 1, the improved recovery methods produce data that are superior to those data computed using Guyan reduction. The data appear to be the most accurate for the simple improvement method. This is especially true when the computer CPU time required to produce the data is considered. The improved displacement recoveries required 30 percent more CPU time than the Guyan recovery, while the super improved displacement recoveries required 150 percent more CPU time than the Guyan recovery.

Because of the simplicity of this model, however, it would be premature to use these data to cast the super improved method aside without first considering more complex models with equally complex loading.

5.0 CONCLUSIONS AND RECOMMENDATIONS

Two methods for improving the O-set displacements were provided. It was demonstrated that both improvement methods produce O-set displacements that are vastly superior to those produced using the standard Guyan recovery alone. In addition, the NASTRAN DMAP ALTERs required to perform these operations were presented along with the supporting data used to verify them. It remains only to determine whether the additional accuracy that may be obtainable through the iterative procedure of Method

3 is justified by the extra computational effort. After all, a significant degree of approximation is already guaranteed by the initial use of Guyan reduction to determine the A-set displacements.

Because this study did not provide enough information to determine which of the two improved recovery methods was best suited for the problems encountered in most engineering applications, it is recommended that additional studies be performed to compare improved displacements from a set of models with varying complexity to the benchmark unreduced data. These additional cases can be used to definitively determine which improvement method is better in terms of accuracy and computational efficiency. Finally, it would be of great interest to compare the results of a multi-mode transient response analysis before and after modal improvement to assess its practical significance in terms of the end result.

6.0 REFERENCES

1. Guyan, R.J., "Reduction of Stiffness and Mass Matrices," AIAA Journal, Volume 3, pg. 380, 1965.

2. Cook, R.D., Concepts and Applications of Finite Element Analysis, John Wiley & Sons Inc., New York, Second Edition, 1981.

3. NASTRAN User's Manual, NASA SP-222, Volume I, June 1986.

4. Ewins, D.J., Modal Testing: Theory and Practice, John Wiley & Sons Inc., New York, June 1985.

MODELLING CHAOTIC VIBRATIONS USING NASTRAN

T. J. SHEERER

CHRYSLER TECHNOLOGIES AIRBORNE SYSTEMS

WACO, TEXAS

ABSTRACT:

Due to the unavailability and, later, prohibitive cost of the computational power required, many phenomena in nonlinear dynamic systems have in the past been addressed in terms of linear systems. Linear systems respond to periodic inputs with periodic outputs, and may be characterised in the time domain or in the frequency domain as convenient. Reduction to the frequency domain is frequently desireable to reduce the amount of computation required for solution.

Nonlinear systems are only soluble in the time domain, and may exhibit a time history which is extremely sensitive to initial conditions. Such systems are termed chaotic.

Dynamic buckling, aeroelasticity, fatigue analysis, control systems and electromechanical actuators are among the areas where chaotic vibrations have been observed. Direct transient analysis over a long time period presents a ready means of simulating the behaviour of self-excited or externally excited nonlinear systems for a range of experimental parameters, either to characterize chaotic behaviour for development of load spectra, or to define its envelope and preclude its occurence.

INTRODUCTION:

Chaotic systems have been defined as those whose time history is highly dependent on initial conditions. Without coining the term "chaos", Henri Poincare (1) informally stated precisely this definition early in the century, and there can be little doubt that earlier than this the concept was known to dynamicists, and remained undeveloped because, in the absence of digital computers and modern instrumentation, it was not a profitable field of inquiry.

The availability of computational power at an unprecedentedly low cost has extended the range of chaotic phenomena in mechanical systems which may profitably be investigated. Such investigation requires solution of the equations of motion of the system in the time domain over a long time period and the subsequent processing of the large body of data acquired to produce phase plots, power spectral densities, peak loads etc. In effect the computer is used to simulate in the time domain a physical test in the time domain (such as a shaker table test for vibration, a wind tunnel test for aeroelasticity, or the experimental observation of the behaviour of an electromechanical system under periodic actuation). Results from the simulation may be processed in the same manner as data from physical experimentation, to produce power spectral densities, Poincare plots and other means of providing insight into the system's behaviour. Extension of analysis beyond the linear domain has the potential of allowing less conservative design assumptions, and of providing an alternative, less statistically oriented approach to load spectrum development and fatigue analysis.

CHAOTIC VIBRATIONS:

Consider a linear dynamic system subject to a periodic input. The response of the system to this input at all degrees of freedom will be a periodic output of amplitude and phase shift dependent upon the mass, stiffness and damping of the system.

The system can be defined equivalently either by equations for displacement as a function of time, or by equations for amplitude and phase of displacement for different input frequencies and amplitudes. The direct response and random analysis disciplines within NASTRAN use the latter approach to generate an output Power Spectral Density (PSD) for a given input PSD to a linear system. Significant modes are determined by modal analysis, after which the amplitude and phase of the system's response to excitation at and around these frequencies using the direct response method. Finally, an input PSD is applied to the data from the direct response analysis to produce an output PSD. of displacement, load, stress or whatever variable is required.

The results obtained are statistical in nature, providing a non-zero value of spectral density for any amplitude. The analyst must determine an amplitude at which nonlinear factors will truncate the PSD curve. This level is somewhat variable, and is generally taken to be between 3 and 10 times the RMS value. Selection of an appropriate truncation point can present problems to the analyst.

Introduction of significantly nonlinear spring constants or nonuniform damping requires that the system must be analysed, in NASTRAN, by direct time integration. Depending upon the degree of nonlinearity and the degree of damping the response to a periodic input may be periodic, quasiperiodic, limit cycle or chaotic. Despite the distinction in names, the first three categories are all periodic in the sense that they may be described by a Fourier series of finite length.

A quasiperiodic system differs from a periodic one in that, although it is expressible as a series of finite length, the frequency components are cannot be expressed as a rational number. It appears, therefore, that quasiperiodic oscillations can not be modelled numerically. Numerical approximation will reduce a quasiperiodic motion to a low frequency periodic one.

Limit cycle vibration is self-excited vibration whose amplitude is limited by non-linear effects. Classical flutter is an example of limit cycle vibration.

Classical flutter theory is limited to the location of regions of negative damping in a linear aeroelastic model, with the purpose of ensuring that these regions are outside the flight envelope. A time-domain solution of nonlinear aeroelatic equations offers the prospect of defining the amplitude of an oscillation which may in reality be either limit cycle or chaotic.

A chaotic system, subject to self-excitation or to a periodic input, will produce a non-periodic output. The system is entirely deterministic and, given the displacement, velocity and acceleration of all degrees of freedom at time t1, the same variables may be calculated at any future time t2. It is interesting to note that the process can not necessarily be reversed to find the state of the system at any prior time. It follows from the above that, if the system is sampled at a rate equal to the period of the input excitation, with any phase shift, the same system state will never recur, since if it did the system would thereafter behave periodically. A self-excited system, not being subject to a periodic external load,

will never exhibit the same state at any sampling frequency.

A useful definition of chaotic vibration might be a response to a periodic input which can not be characterised by a Fourier series of finite length.

Time-domain analysis of potentially chaotic vibrations subject to periodic excitation provides information as to range of frequencies and amplitudes of excitation for which a non-periodic response may be expected, by examination of power spectral density and Poincare plots, and also information allowing an informed decision as to where to truncate the output PSD from a random response analysis, if the response should prove to be approximately linear for the levels of excitation of interest. For systems where the excitation is dominated by a relatively small number of frequencies, the system can be solved directly over a suitable time period by using a combination of dynamic load cards to provide excitation with several frequency components. Input excitations associated with rotating machinery are a case in point.

In self-excited oscillations, such as flutter, a non-linear analysis in the time domain can, by accounting for geometic and material nonlinearities, provide the limit amplitude of a periodic oscillation, or an envelope for chaotic oscillation. Other potentially chaotic self-excited systems include control systems with hysteresis and "galloping" of cables.

In all these cases it is potentially of interest to determine whether the oscillation will result in immediate catastrophic failure, will produce stresses affecting the life of the structure or will be limited at a benign level by nonliearities.

ATTRACTORS, POINCARE MAPS AND POWER SPECTRAL DENSITY

Given a time history of a time-domain NASTRAN transient analysis, of a self- or periodically excited system, the generation of an output PSD is an obvious and simple step. This involves operating on the output data in precisely the same manner as would be done with experimental data. At least as important for potentially chaotic systems are phase plots and Poincare plots, where the variable of interest (usually position) is the ordinate and its first derivative is the abscissa.

For a periodic oscillation, either externally or self-excited, such a plot will form a closed path. The simplest case, an undamped single DOF oscillator, appears in a phase plot as an ellipse (or a circle if appropriately scaled) centered on the equilibrium position of the oscillat of a damping term will produce a plot in phase space which spirals in to t equilibrium position. The equilibrium point is an attractor for the single DOF damped spring, since as the initial disturbance of the system dies away, the system tends to this state. For a periodic oscillation not decaying to equilibrium, such as the undamped single DOF oscillator, the attractor is a close curve. Sampling at a rate equal to the natural frequency will reduce the plot to a single point. Such plots in phase spac are termed Poincare plots. More complex periodic oscillations, having several frequency components due to a forcing function with several freque will appear in the phase plot as interleaved curves. By selecting a sampli the appropriate sampling rate the output data will be a finite number of loci defining a closed curve, with data points repeating after a finite number of cycles. In a single DOF system, sampled at the forcing function frequency, the coincidence of displacement and velocity implies a coincide of acceleration, and consequently the curve in the phase plot can not inte itself.

For a quasiperiodic oscillation the attractor will form a closed curve sampled at in phase space. Although all points will lie on the curve, none

Results of the analysis may be interpreted in the same way as those of a physical test.

(1): A time history of displacement or velocity may exhibit a clear periodicity or may not. In the latter case the cause could be either chaotic motion or the combination of several periodic components.

(2): Power Spectral Density Analysis of the system response to a single frequency forcing function. A system verging upon chaos will exhibit several harmonics of the driving frequency, with the response becoming broad-band as the system enters the chaotic regime. Judgement as to the presence or absence of chaos must be made with regard to the system analysed. In the case analysed below a single DOF system produces several harmonics for certain levels of periodic excitation. The conclusions drawn from it would not necessarily be justified from observations of a single node in a complex structure.

(3,4,5): Phase plane observation, Poincare and 3-D plots: These are discussed in some detail above.

DYNAMIC MODEL OF AN ELECTROMECHANICAL ACTUATOR SYSTEM:

The electromechanical actuator is a known, simple example of a chaotic oscillator, described by Hendricks in 1983 (2).

Fig.(1) shows an electromechanical actuator system wherein the armature is subject to an externally applied dynamic load by application of an electrical current to a coil. Such systems are used in impact print mechanisms, high speed relays and elsewhere.

The system is modelled as an armature GRID with a single DOF moving between two GRIDS each occupying a deep potential well defined by NOLIN1 cards and representing the stops limiting the armature's travel. EPOINT NOLIN1 and TF cards are used to model the impacting of the armature on the stops.

The armature GRID is also attached to ground by a scalar spring whose stiffness was varied during the investigation. The armature thus tends to a rest position with the scalar spring in an unloaded state as shown in Fig.(2).

Also in Fig.(1) is a mechanical fastener transfering load between two components having oversized holes. This system, representative of structural details in aircraft construction or modification, is from a mathematical point of view identical with the actuator system. Note that by applying excitation at one of the constraining grids the same model can represent, without further modification, a system with nonlinear stiffness mounted on a a shaker table.

The actuator modelled was given travel between stops of 0.008 inch, peak applied force of 0.8 # and cycle time of 1KHz. These times were based upon an actual device for which data was available and were varied in the course of the study to induce chaotic behaviour. It was determined that a time step of 0.5 uS was required to adequately model the behaviour of the armature and stops during impact. Inspection of the motion of the stops shows that they are restored to equilibrium position between impacts and hence act merely as nonlinear restoring forces on the armature. The armature therefore acted, in effect, as a single DOF nonlinear system. A means of applying a load as a function of space and time was also devised and is described in appendix (1).

RESULTS:

(1): VARIATION OF DYNAMIC LOAD

Curves of displacement vs. time are plotted in Figs. (3-7) for excitation at 1kHz with peak forces from 0.1 # to 1.2 #, with a travel of 0.008 inch between stops and a weak spring defining the rest position of the armature. It is seen that for the extreme limits of applied load the results do not appear to be periodic.

will be coincident since the ratio of the component frequencies is not a rational number. In a time-domain simulation the distinction from a perioc oscillation is of no importance.

A chaotic oscillation, sampled in this manner, will never repeat itself and may exhibit an interleaved phase plot. This state, not conforming to any of the three cases in classical dynamics, is termed a strange attracto While the static, periodic and quasiperiodic attractors define closed path strange attractors, while being confined to a finite area of phase space, exhibit fine structure within their domain. Alternatively, in lightly damped systems, the plot may appear to be randomly distributed. Such systems are sometimes described as stochastic in nature.

A plot of displacement, velocity and acceleration is of interest. In a self-excited single DOF system, the coincidence of position and velocity imply a coincidence of acceleration, since the acceleration is defined in terms of the other two variables. In chaotic systems, the converse is true and no two points may be coincident in such a plot. In a system with several degrees of freedom the presence or absence of periodicity must be determined by examining, and seeking a coincidence in, the displacement and velocity of all degrees of freedom simultaneously. Graphically, this requires plotting in a space of 2N dimensions where N is the number of degrees of freedom. For a system subject to an external forcing function, the sampling must be done at the frequency of the forcin function. Given that the analysis must be based upon a simulation of finit span, it will not be possible to prove explicitly that a system is chaotic and only in some clear-cut cases will it be possible to prove the converse

In practice, as in actual physical testing, several tests may be applie which with a high degree of confidence discriminate between chaotic and periodic behaviour. The envelope defined for motion of an apparently chaotic system is no less useful if the system in fact is periodic with a very long wavelength.

APPLICATION OF NASTRAN TRANSIENT ANALYSIS

The paradigm of chaos, the Lorenz attractor, was initially attributed by some to the process of numerical simulation rather than to an underlyin physical reality. This proposition will be sympathetically viewed by any analyst who has used NASTRAN to model intermittent contact problems.

In impact studies and similar applications the greatest care must be ta to ensure that the time step is sufficiently small to prevent a node from penetrating a significant distance into a region of high stiffness before the stiffness matrix is updated to reflect this. The effect of such an excessive time step can be that the node is reflected from the collision with a velocity many times that of impact. At the same time, the total number of time steps must, as far as possible, be minimized. For problems such as a single impact, where the regions requiring small timesteps can be estimated, or derived from a preliminary analysis, the problem may be addressed by using several timesteps, with the small ones limited to the appropriate times. In analysis of a chaotic system, however, a large number of cycles must be analysed, and the behaviour is by definition nonperiodic and unpredictable. A single value of time step must be employe and experimentation is required to determine the maximum timestep commensurate with conservation of energy in the system.

The application of periodic dynamic loads required the input of a large amount of data, defining each of many cycles explicitly. This is conveniently done using an external preprocessor to generate the appropriate cards. The numerical and graphical output from direct transie consists of the values of variables as a function of time, as would be the case for a physical test. The desired output of phase plots (velocity vs. position) and power spectral density may be readily obtained, however, fro a punch file of the results, either by use of a batch program or by importation into a spreadsheet or database program.

The velocity plots in Figs. (8-12) provide a clearer picture of the armature's behaviour, with almost constant peak velocity for dynamic loads from 0.24 to 0.8 # and considerable variation outside that envelope. Figs. (13-17) are phase plots of velocity vs. displacement for the same data. Figs. (18-19) show the superimposed plots in the vicinity of the front and back stops. The larger scale reveals considerable fine structure in the curves for peak dynamic loads from 0.24 # to 0.8 #.

Fig.(20) shows the displacement PSD for peak dynamic loads from 0.24 # to 1.2 #. The largest peak is for the 0.4# peak force, but, if the results are normalised for the amplitude of the input force, the 0.24# case will have almost the same magnitude, but with much less marked secondary peaks.

Fig.(21) shows the 0.4# and 0.24# Poincare plots for a sampling rate twice the excitation rate, phase shifted to encompass maximum deflection. The loci near the equilibrium attractor are virtually coincident while the loci mear maximum displacement show considerable variation in velocity, but not position. Fig.(22) shows loci for peak forces of 1.2# and 0.24# for a sampling rate equal to the excitation rate. The 0.24# case suggests a long-period periodicity while the 1.2 # case suggests chaotic vibration. Figs. (23-24) are 3-D plots for peak loads of 0.8 # and 1.2 # respectively.

(2): VARIATION OF NONLINEARITY

By increase of the linear spring constant constraining the armature from 1.0# to 100.0 # it becomes significant with respect to the nonlinear forces. Fig. (25) shows the displacement vs. time for a peak dynamic load 0f 0.8 # for spring constants of 1.0 and 100.0 respectively. It is apparent from this and the Poincare plot in Fig. (26) that the effect of reducing the range of stiffness is to reduce the tendency to chaos.

(3): VARIATION ON INPUT FREQUENCY

The effect of increasing input frequency is to increase the tendency to chaos. Fig.(27) shows phase plots for 0.8# peak input force at 1.0, 1.5 and 2.0 Khz. The chaotic behavior at 2.0 KHz is in accordance with test data indicating a maximum stable drive frequency around 1.7 kHz.

CONCLUSIONS:

The data described above for a magnetomechanical actuator are in agreement with several years of experience in the design, analysis and characterization of such devices. With small modifications, a similar model could be applied to mechanical fasteners in aircraft structures, vibration isolation and other areas where load transmission between pieces of structure is via a nonlinear path. Application of appropriate position-dependent loads should allow nonlinear modelling of flutter and other self-excited phenomena.

Considerable care must be taken to ensure that effects observed are due to physical characteristics of the system and not artifacts of the simulation. Spurious self-excitation of the system due to an inadequate time step is an obvious possibility.

Implementation of automatic time-step variation, such as is available in some other FEA codes, is probably not desireable for an application where there is a significant risk of mistaking numerical artifacts for physical behaviour. A means of specifying a large number of periodic loads on a single card would, however, be desireable.

REFERENCES:

(1): H. Poincare, The Foundation of Science: Science and Method (1921)

(2): F. Hendricks, Bounce and Chaotic Motion in Print Hammers, IBM J Res. Dev., 27(3), 273-280

Fig. (1): 1-D Nonlinear System

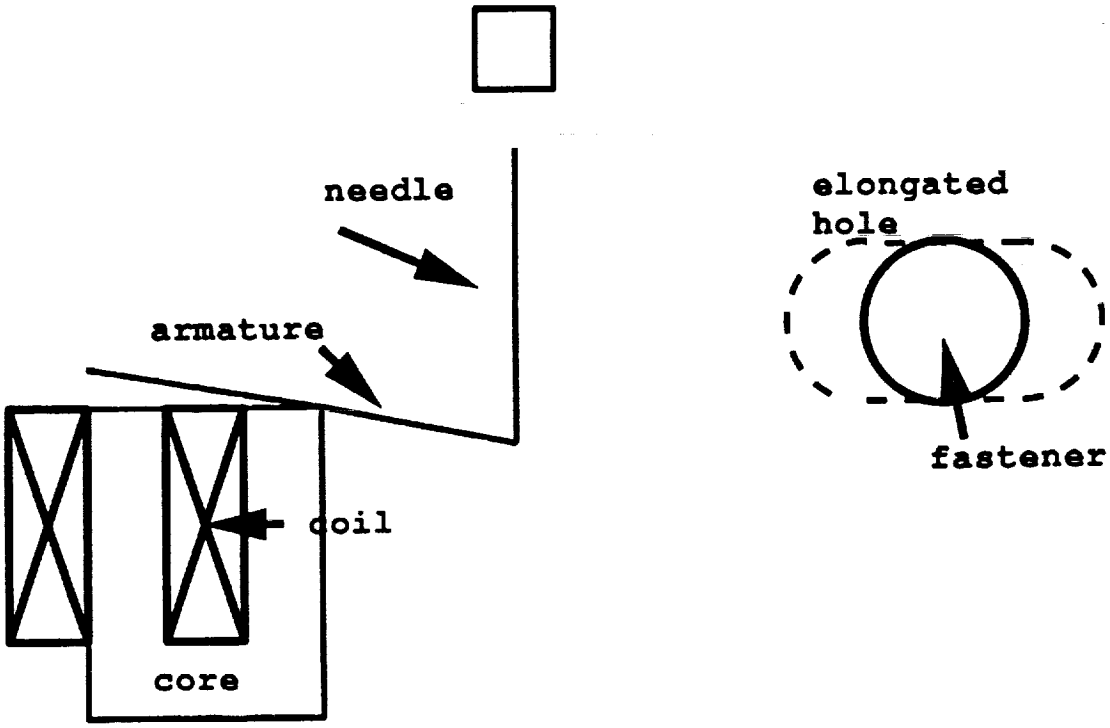
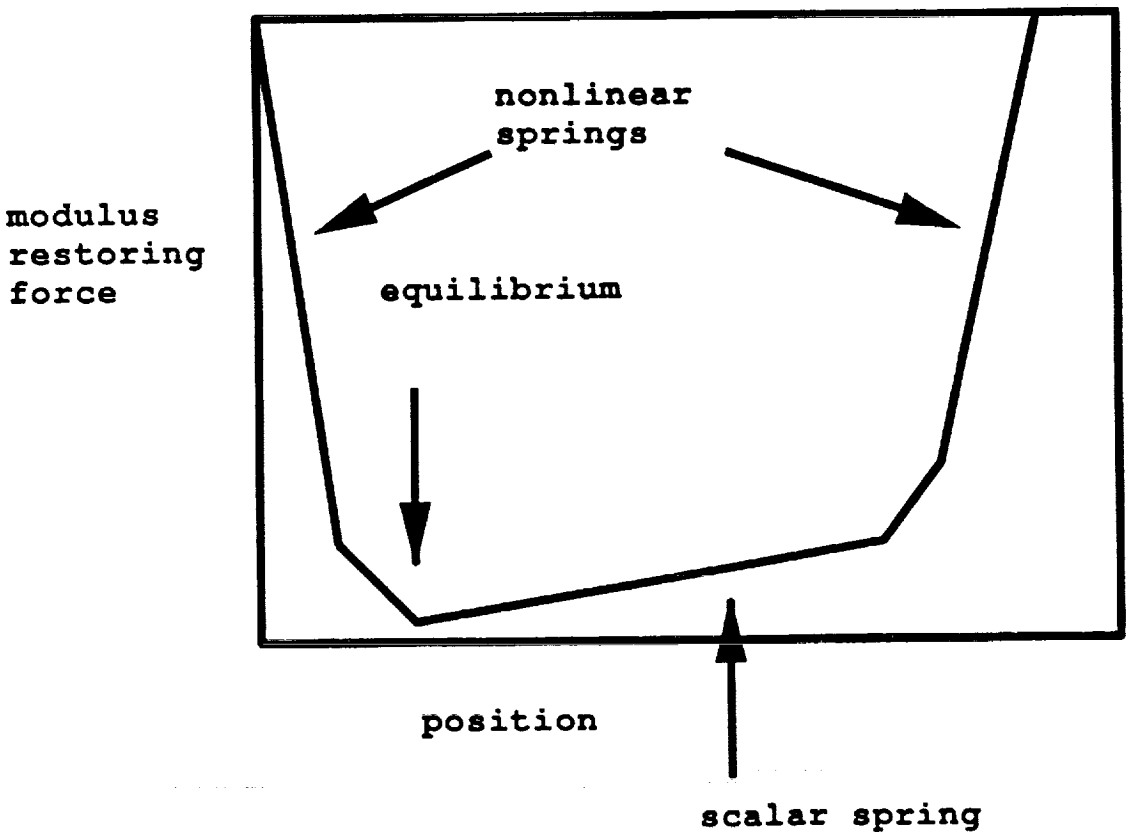
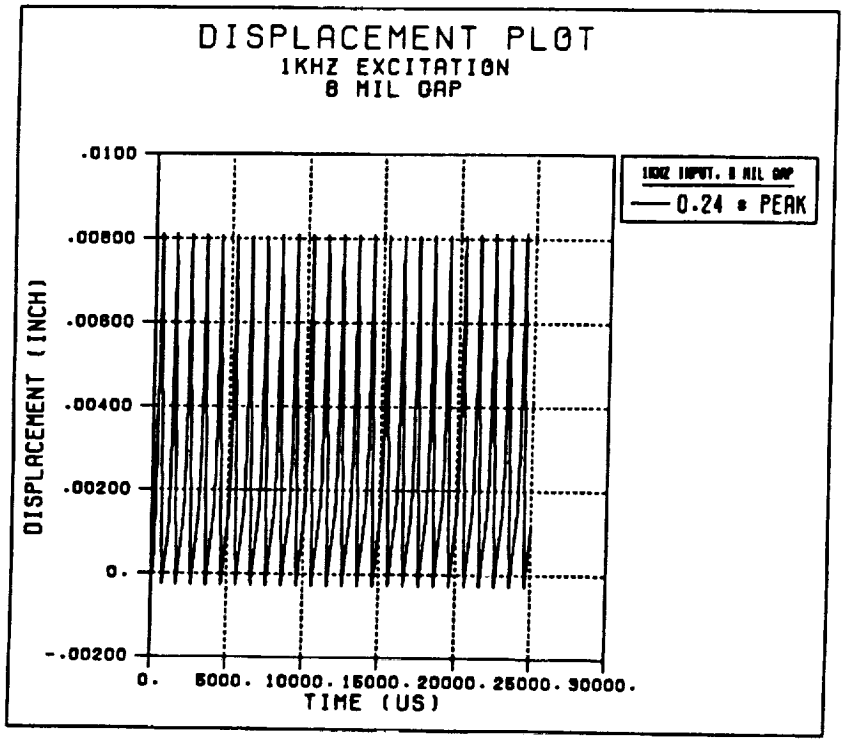
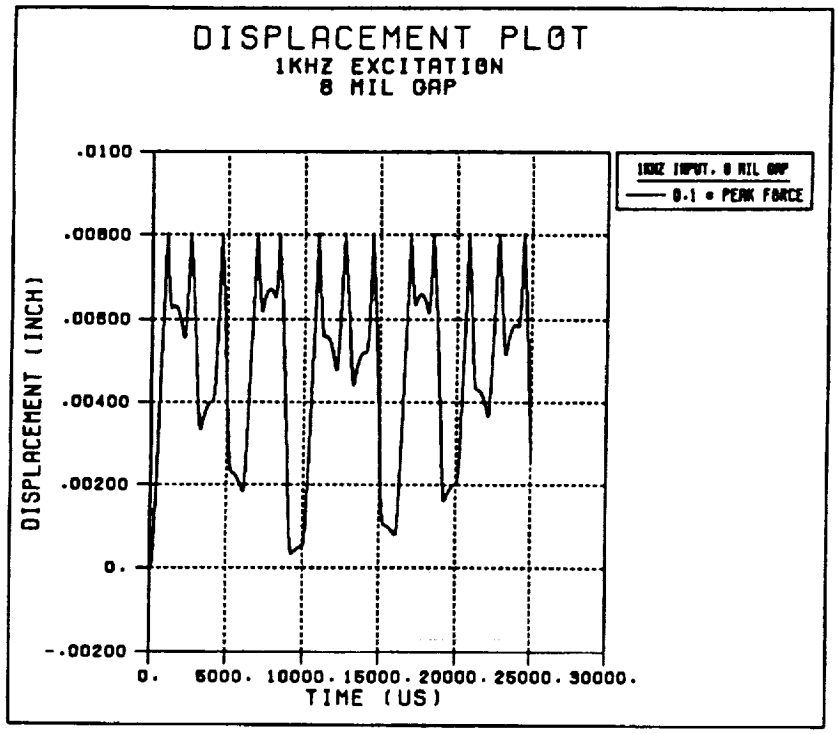


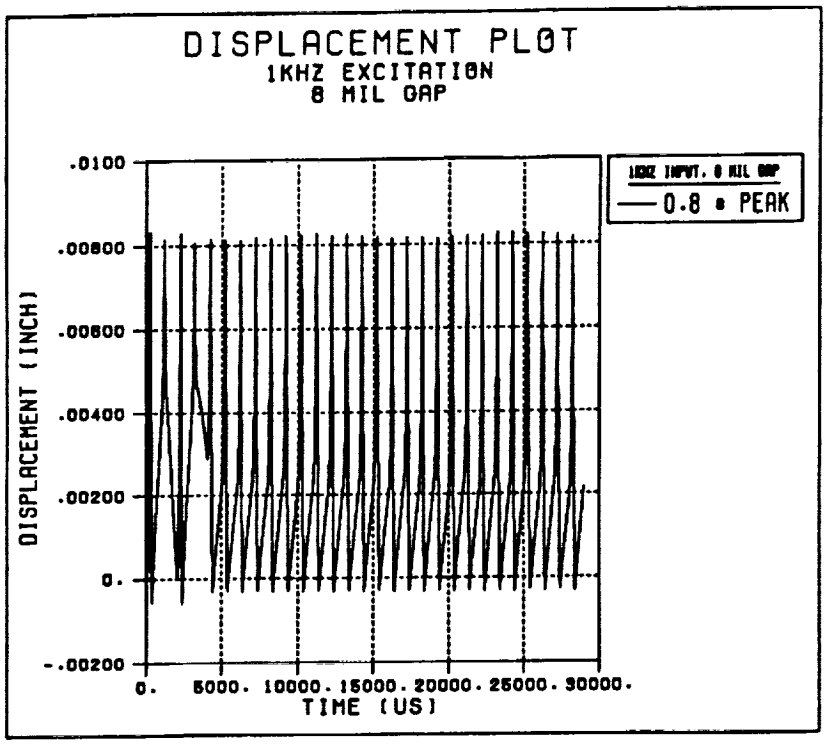
Fig. (2): Restoring force in system

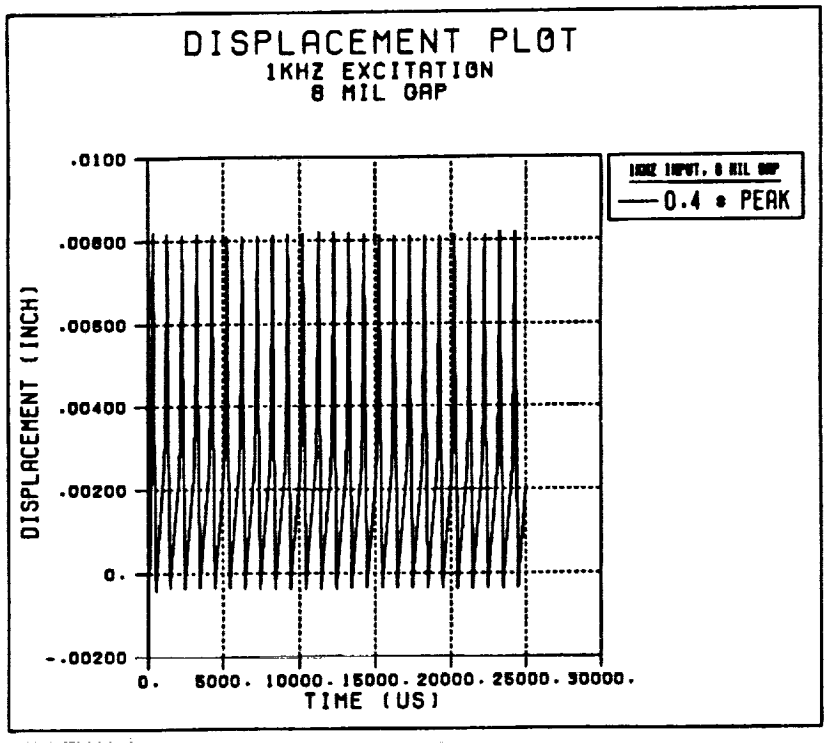


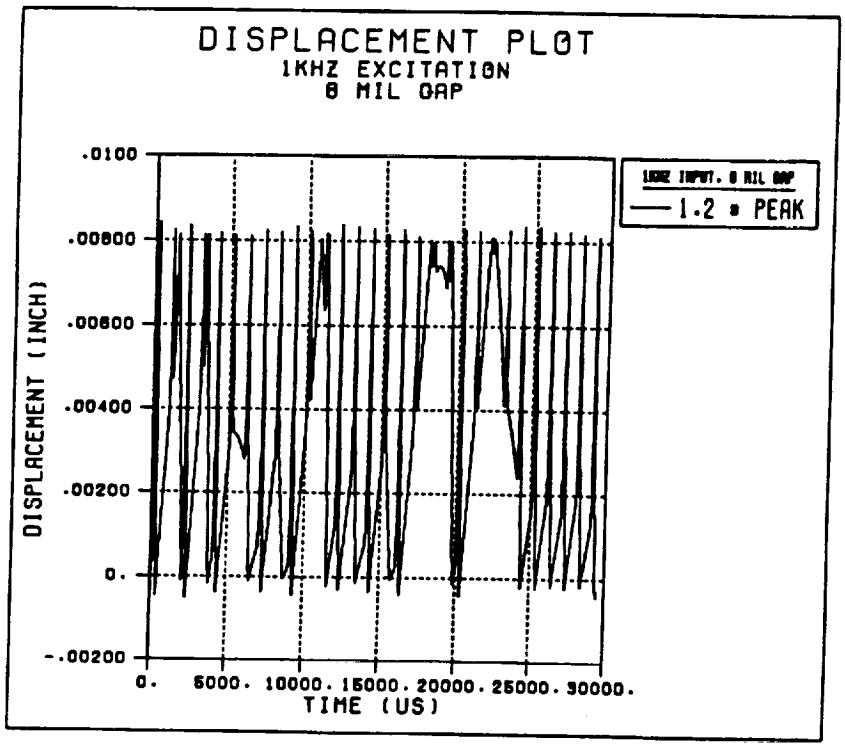
Figs (3-7): Displacement vs. time for armature



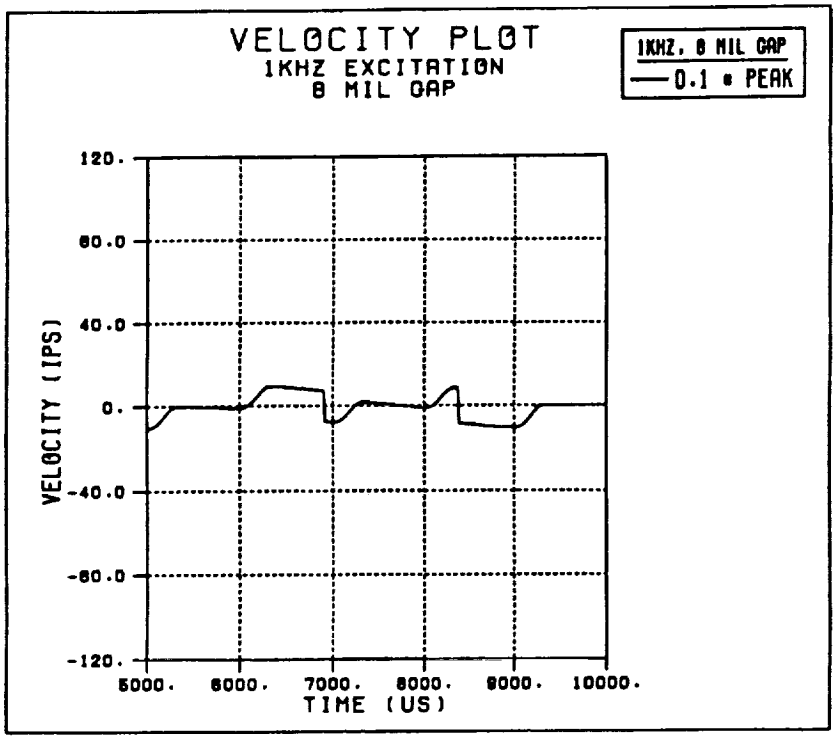


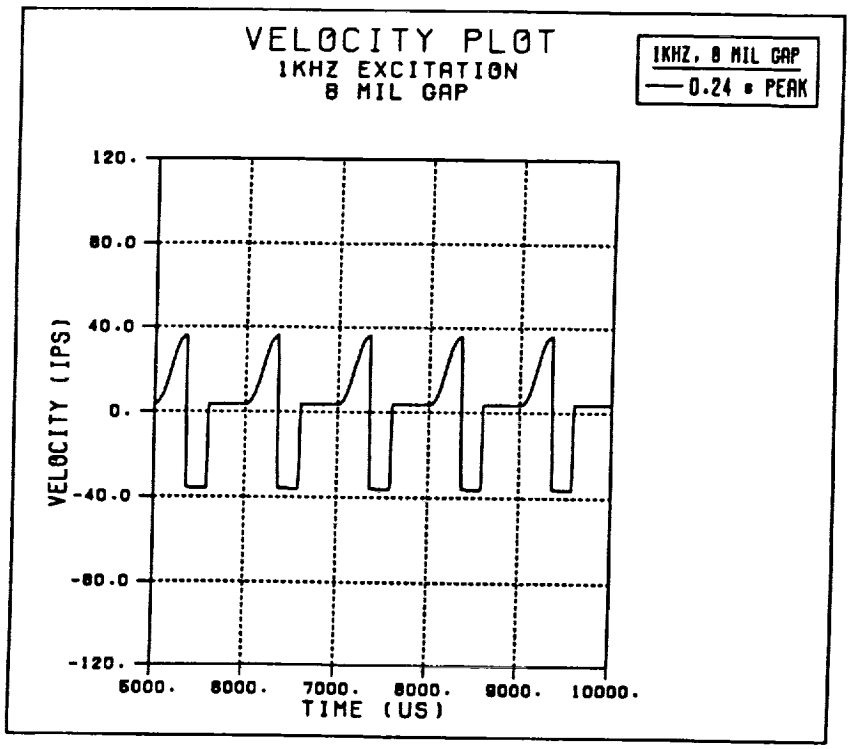

DISPLACEMENT PLOT
1KHZ EXCITATION
8 MIL GAP
.0100
.00800
.00600
.00400
.00200
0.
-.00200
DISPLACEMENT (INCH)
0.
5000.
10000.
15000.
20000.
25000.
30000.
TIME (US)
0.8 * PEAK

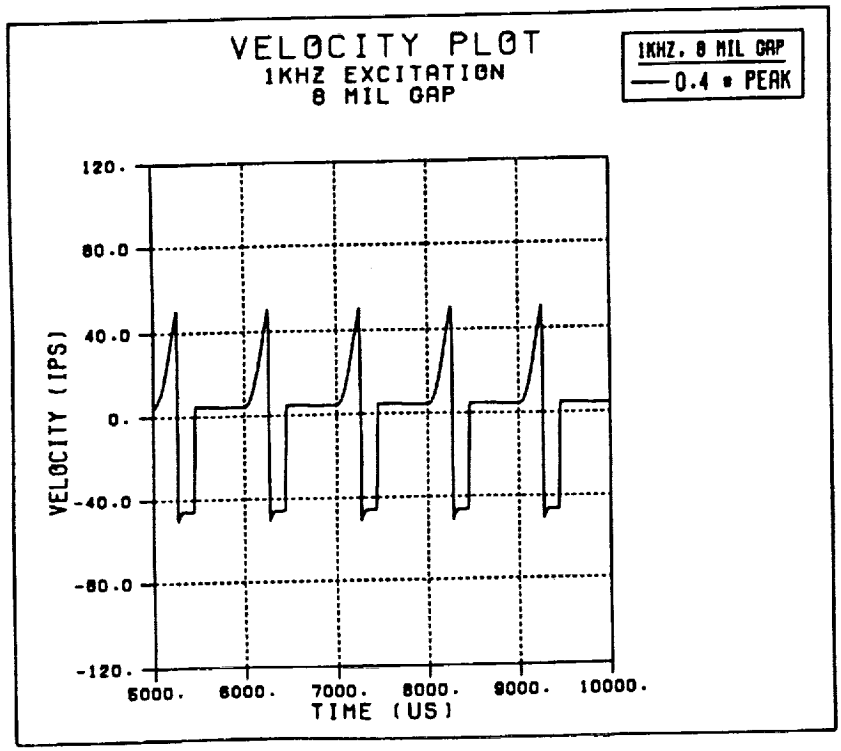

DISPLACEMENT PLOT
1KHZ EXCITATION
8 MIL GAP
.0100
.00800
.00600
.00400
.00200
0.
-.00200
DISPLACEMENT (INCH)
0.
5000.
10000.
15000.
20000.
25000.
30000.
TIME (US)
0.4 * PEAK

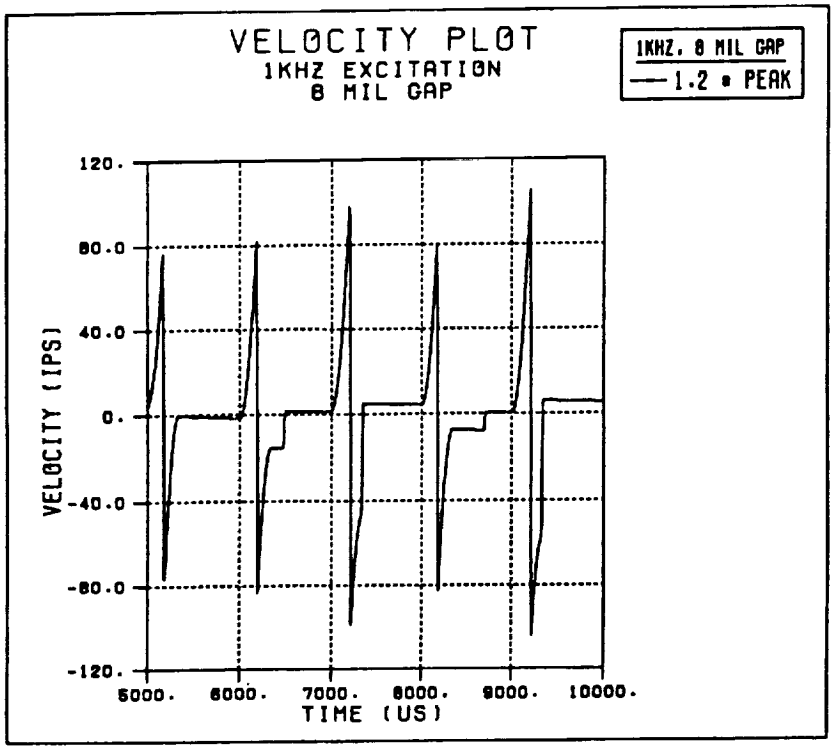


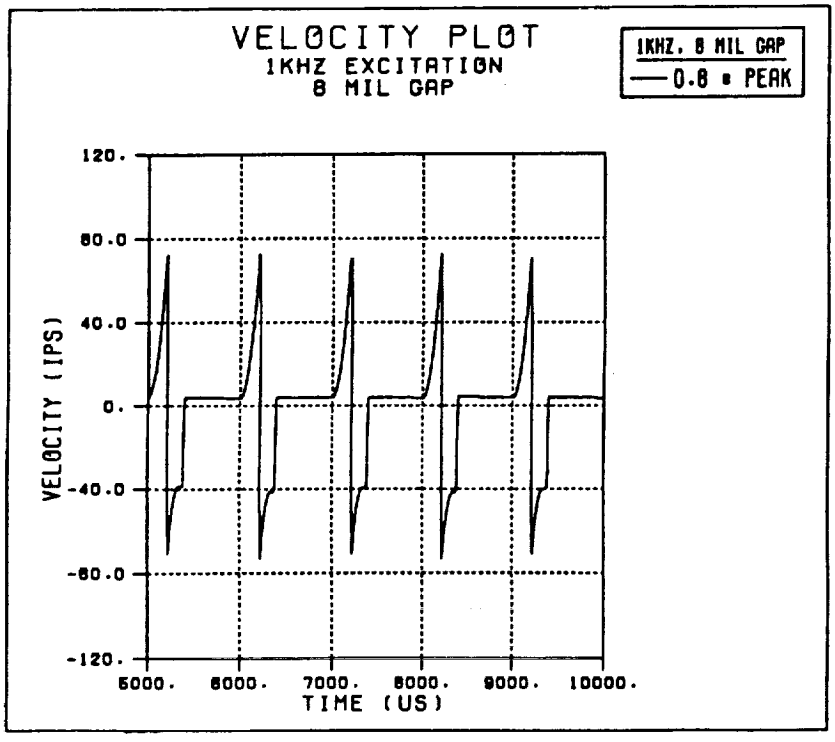
Figs. (8-12): velocity vs. time for armature



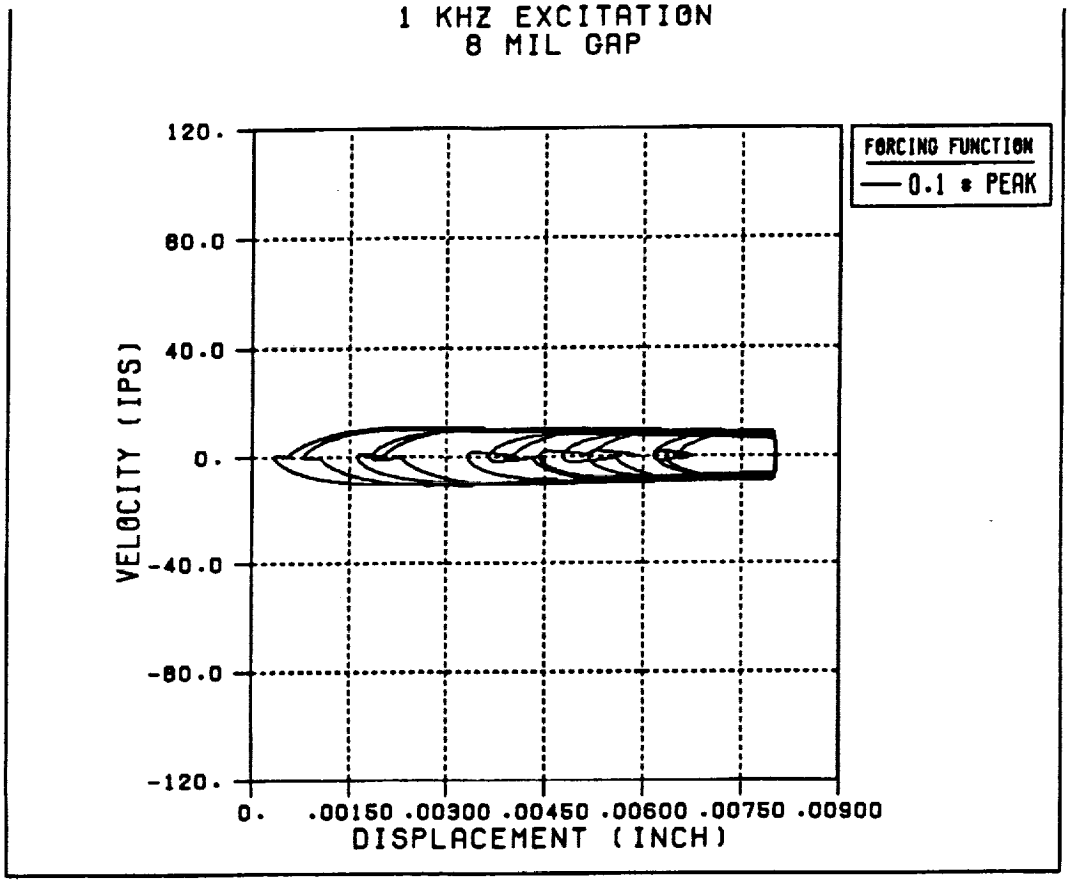

VELOCITY PLOT
1KHZ EXCITATION
8 MIL GAP
1KHZ, 8 MIL GAP
0.24 * PEAK
120.
80.0
40.0
0.
-40.0
-80.0
-120.
VELOCITY (IPS)
5000.
6000.
7000.
8000.
9000.
10000.
TIME (US)

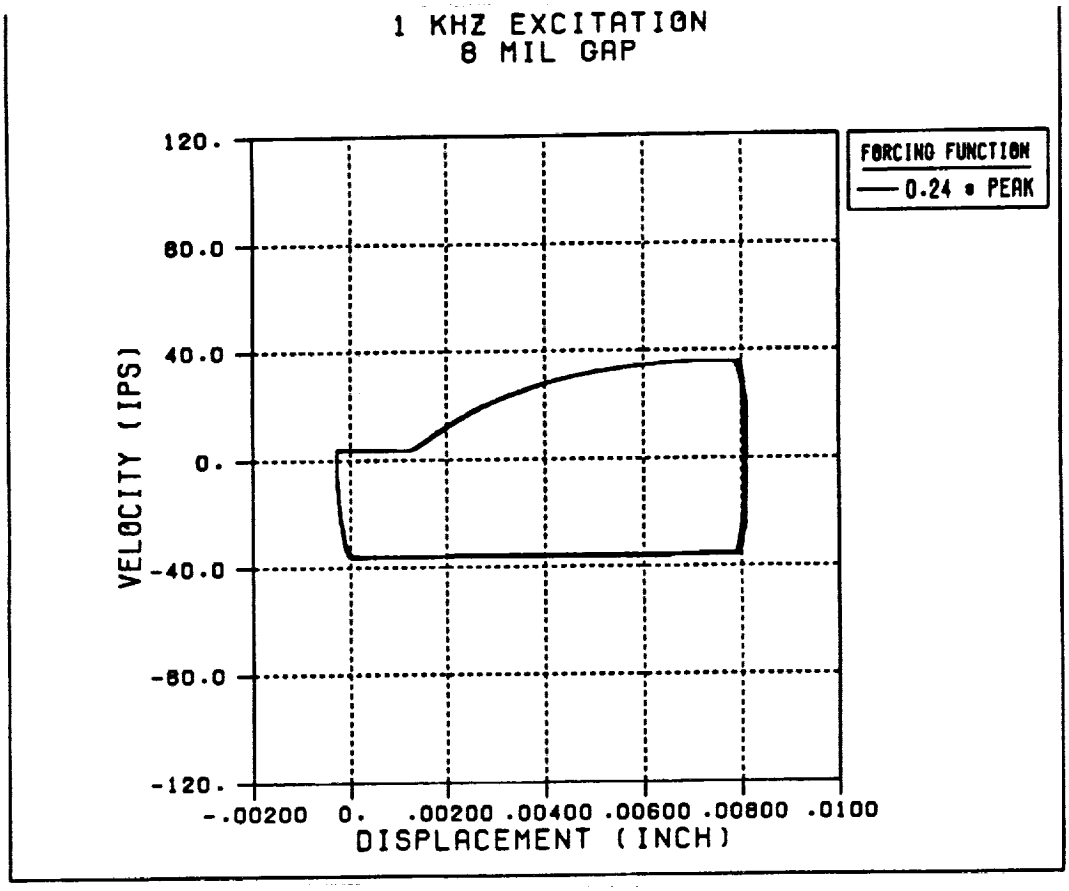

VELOCITY PLOT
1KHZ EXCITATION
8 MIL GAP
1KHZ, 8 MIL GAP
0.4 * PEAK
120.
80.0
40.0
0.
-40.0
-80.0
-120.
VELOCITY (IPS)
5000.
6000.
7000.
8000.
9000.
10000.
TIME (US)

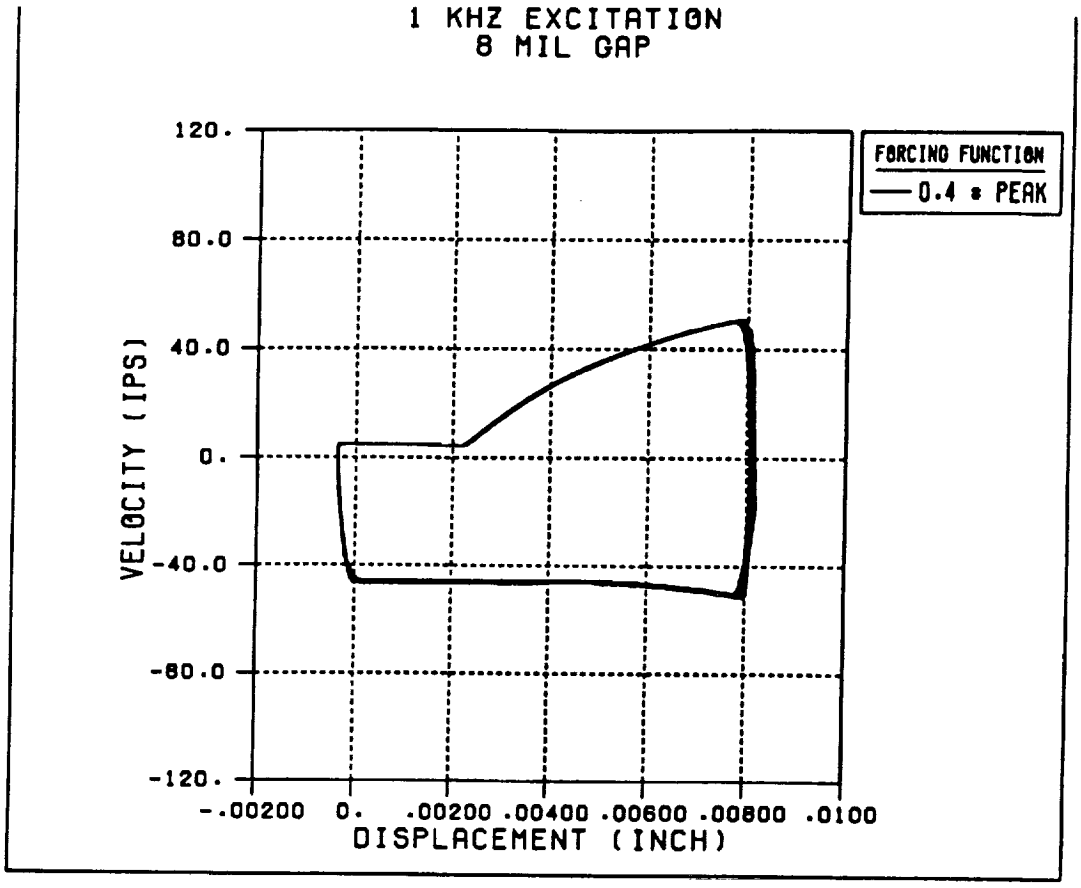

VELOCITY PLOT
1KHZ EXCITATION
8 MIL GAP
1KHZ, 8 MIL GAP
1.2 * PEAK
120.
80.0
40.0
0.
-40.0
-80.0
-120.
VELOCITY (IPS)
5000.
6000.
7000.
8000.
9000.
10000.
TIME (US)

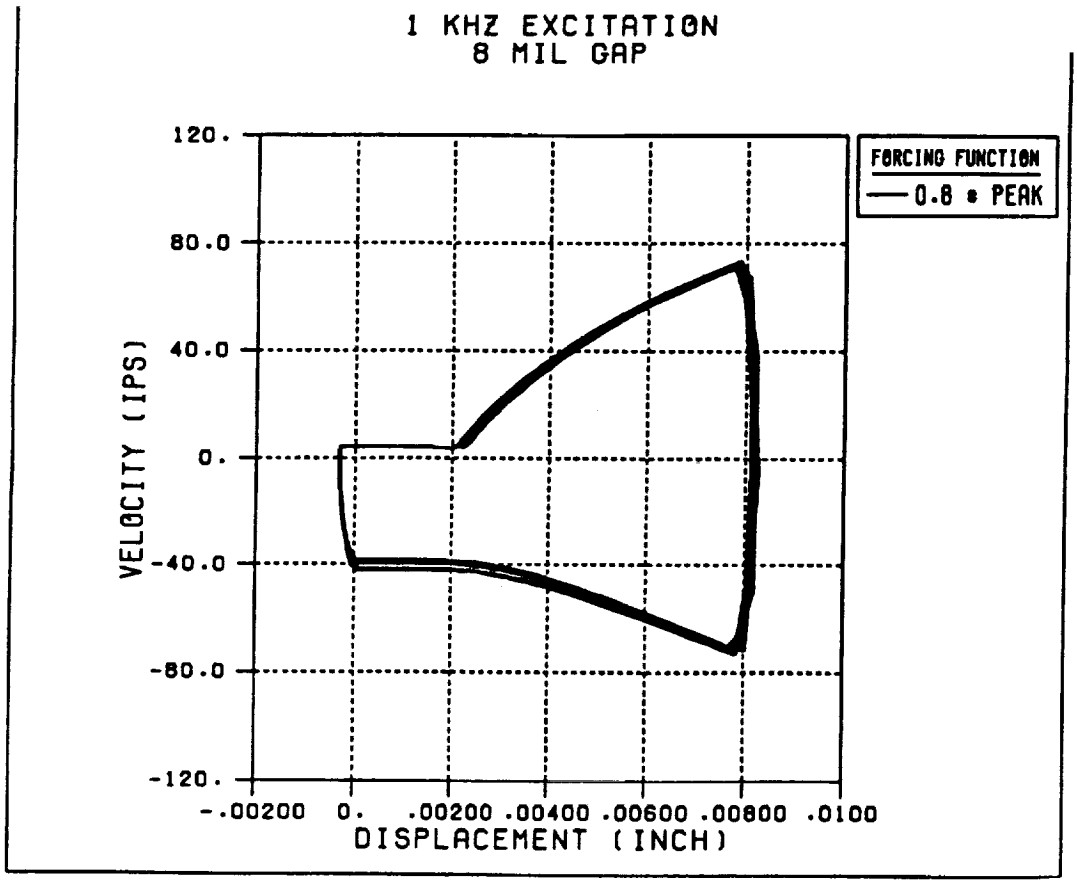

VELOCITY PLOT
1KHZ EXCITATION
8 MIL GAP
1KHZ, 8 MIL GAP
0.8 * PEAK
120.
80.0
40.0
0.
-40.0
-80.0
-120.
VELOCITY (IPS)
5000.
6000.
7000.
8000.
9000.
10000.
TIME (US)

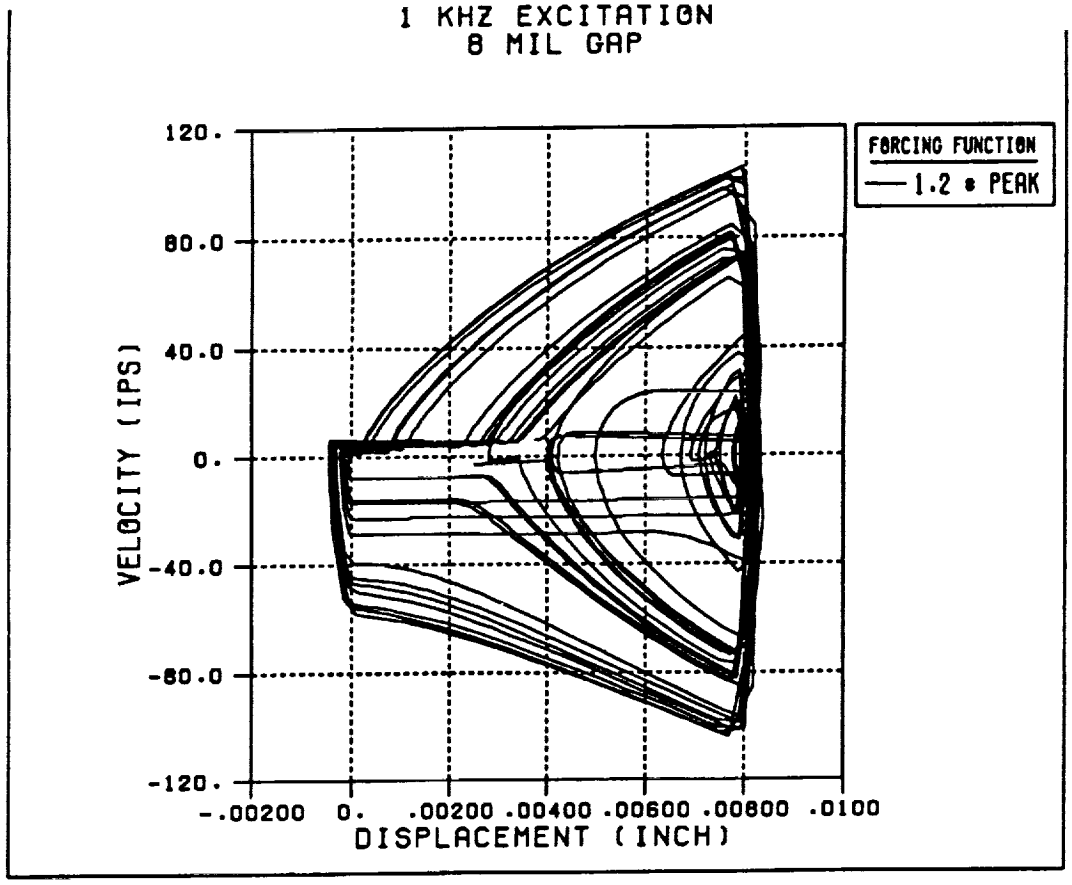
Figs. (13-19): velocity vs. displacement

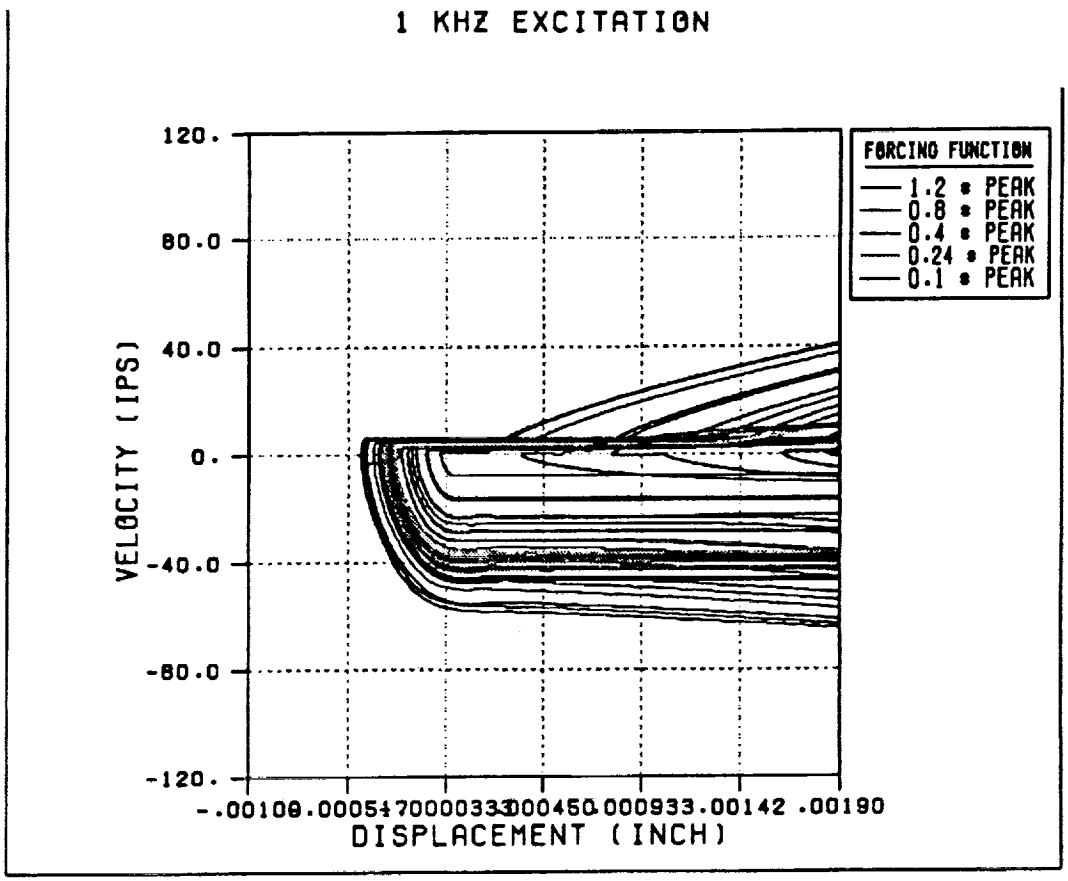





1 KHZ EXCITATION
8 MIL GAP
FORCING FUNCTION
0.4 * PEAK
120.
80.0
40.0
0.
-40.0
-80.0
-120.
VELOCITY (IPS)
-.00200
0.
.00200
.00400
.00600
.00800
.0100
DISPLACEMENT (INCH)


1 KHZ EXCITATION
8 MIL GAP
FORCING FUNCTION
0.8 * PEAK
120.
80.0
40.0
0.
-40.0
-80.0
-120.
VELOCITY (IPS)
-.00200
0.
.00200
.00400
.00600
.00800
.0100
DISPLACEMENT (INCH)


1 KHZ EXCITATION
8 MIL GAP
FORCING FUNCTION
1.2 * PEAK
120.
80.0
40.0
0.
-40.0
-80.0
-120.
VELOCITY (IPS)
-.00200
0.
.00200
.00400
.00600
.00800
.0100
DISPLACEMENT (INCH)


1 KHZ EXCITATION
FORCING FUNCTION
1.2 * PEAK
0.8 * PEAK
0.4 * PEAK
0.24 * PEAK
0.1 * PEAK
120.
80.0
40.0
0.
-40.0
-80.0
-120.
VELOCITY (IPS)
-.00190
.00142
DISPLACEMENT (INCH)

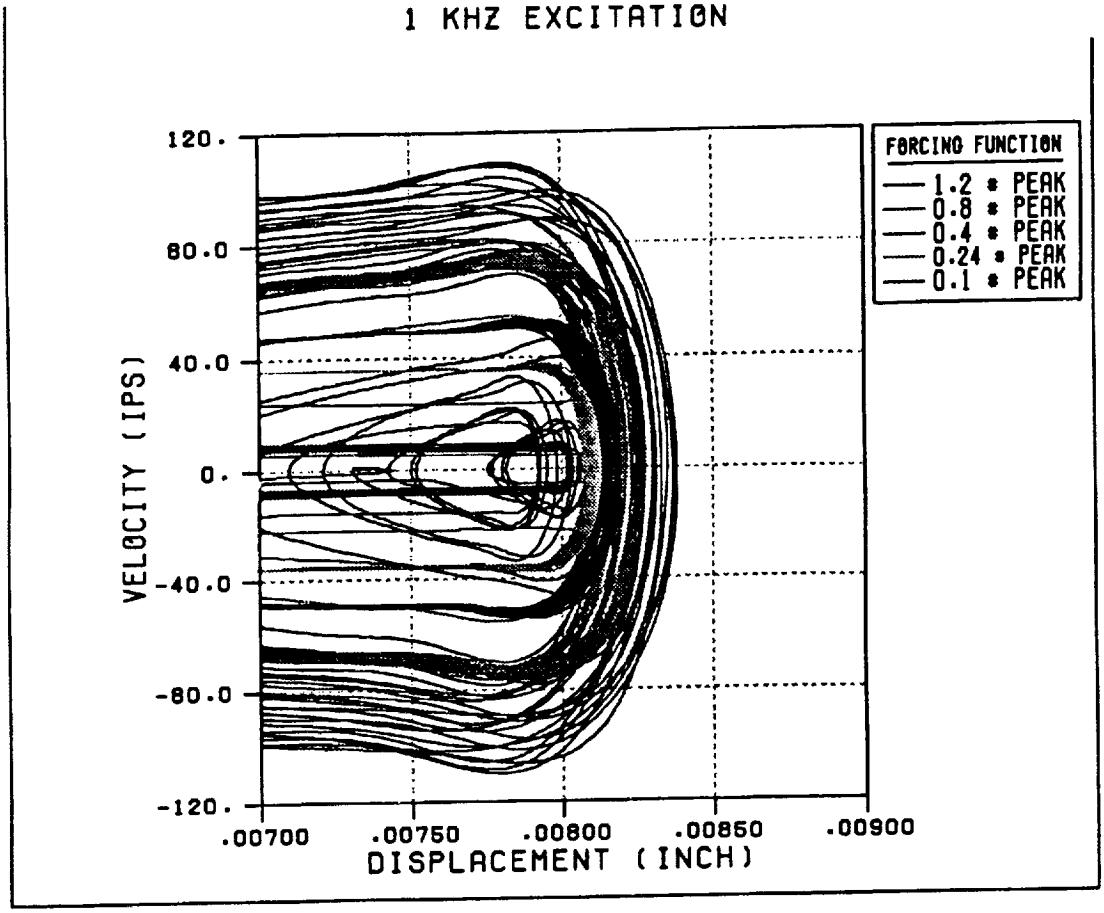
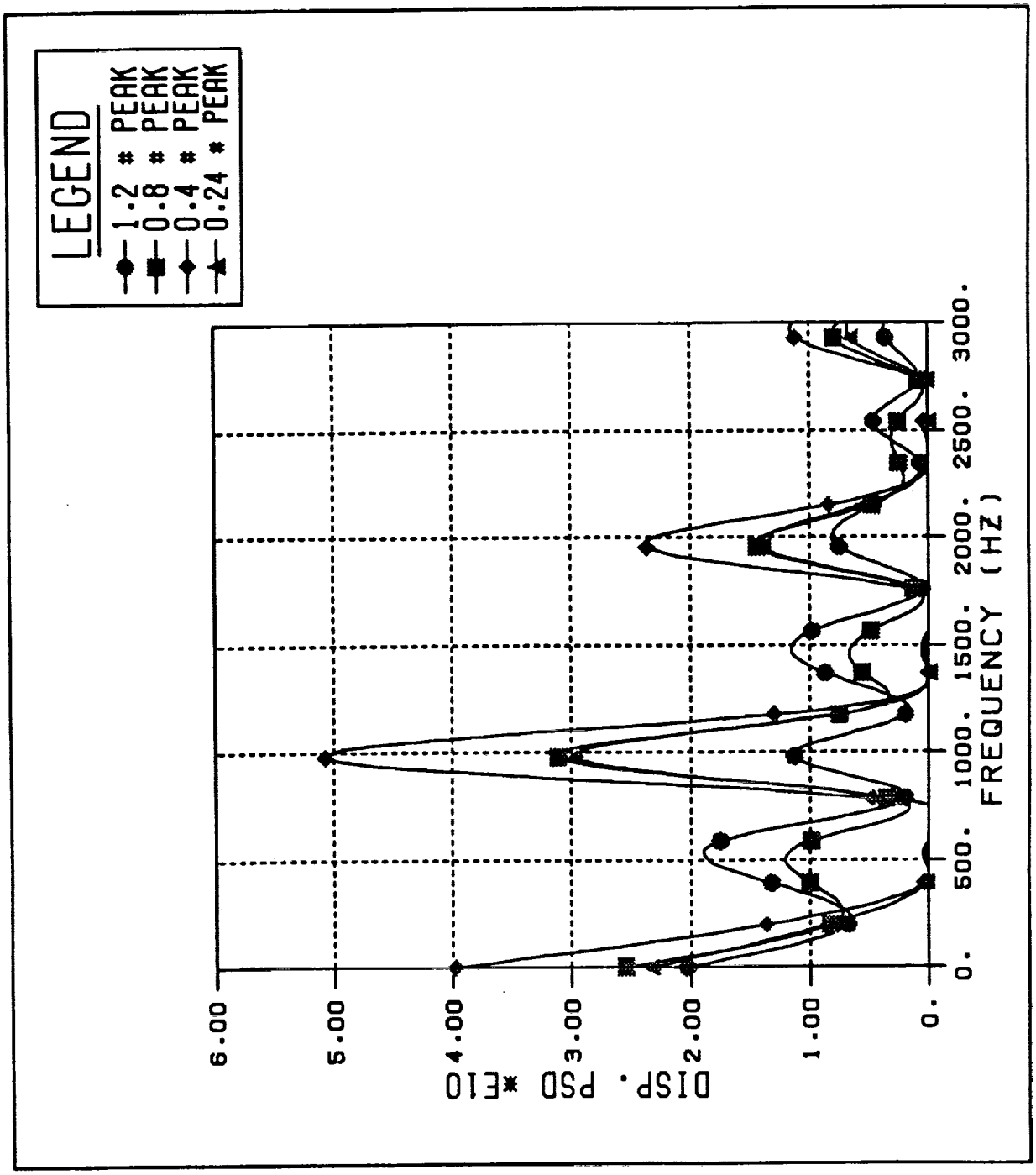
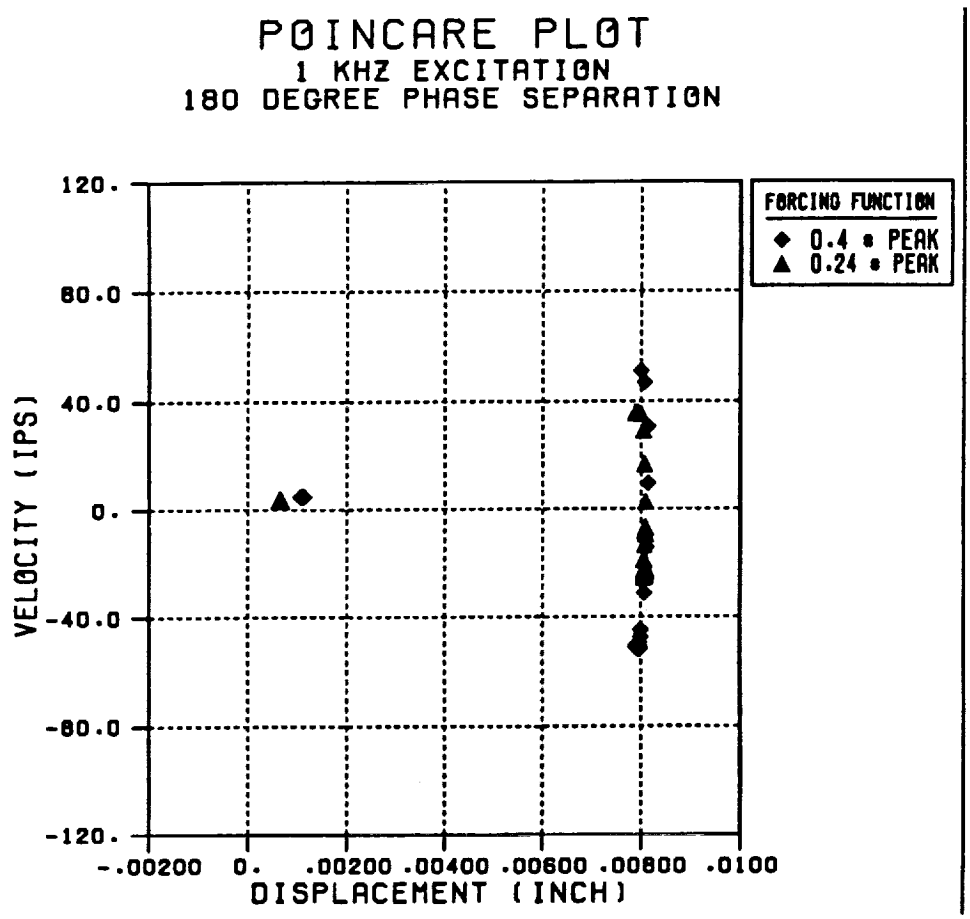

1 KHZ EXCITATION
120.
80.0
40.0
0.
-40.0
-80.0
-120.
VELOCITY (IPS)
.00700
.00750
.00800
.00850
.00900
DISPLACEMENT (INCH)
FORCING FUNCTION
1.2 * PEAK
0.8 * PEAK
0.4 * PEAK
0.24 * PEAK
0.1 * PEAK

Fig. (20): Displacement Power Spectral Density



Figs (21-22), Poincare plots sampled at 2kHz



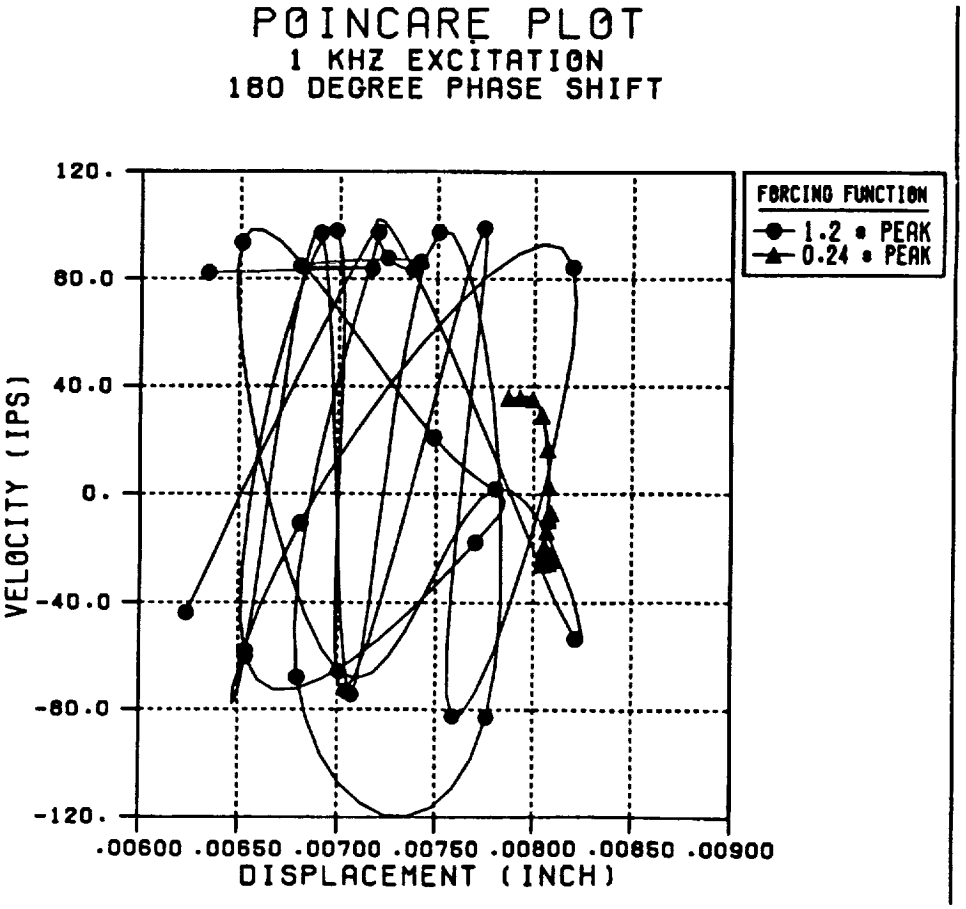


Fig. (23): 3-D plot for 0.8 # peak load

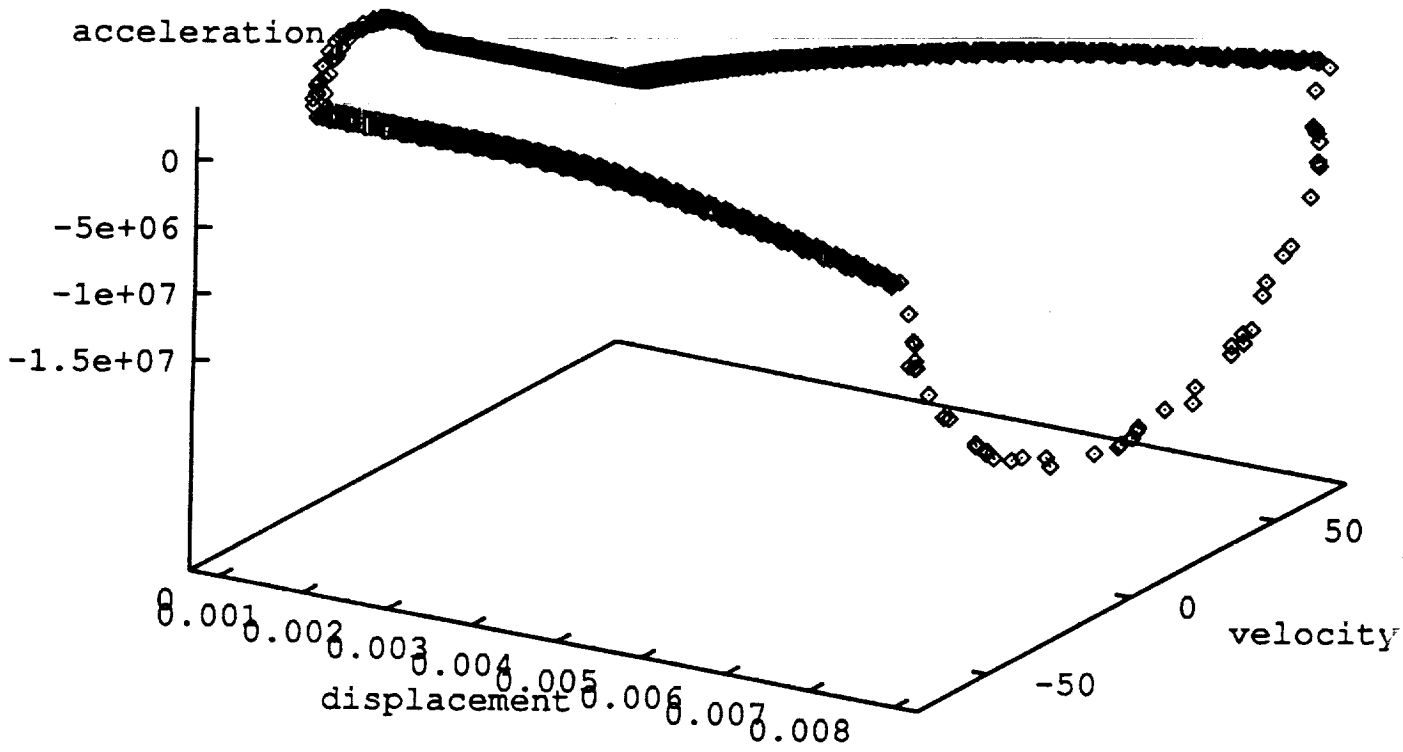


Fig. (24): 3-D plot for 1.2 # peak load

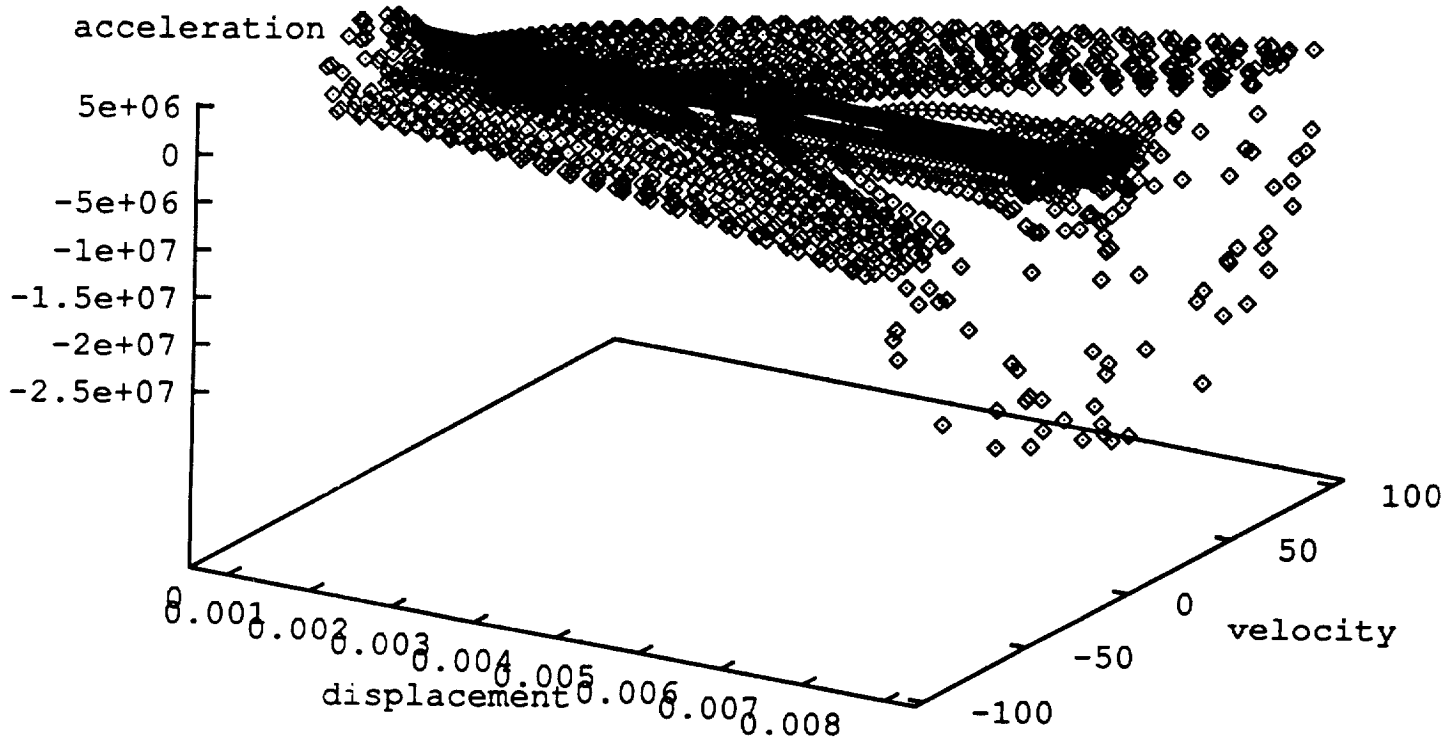


Fig. (25): displacement plots for different linear spring constants

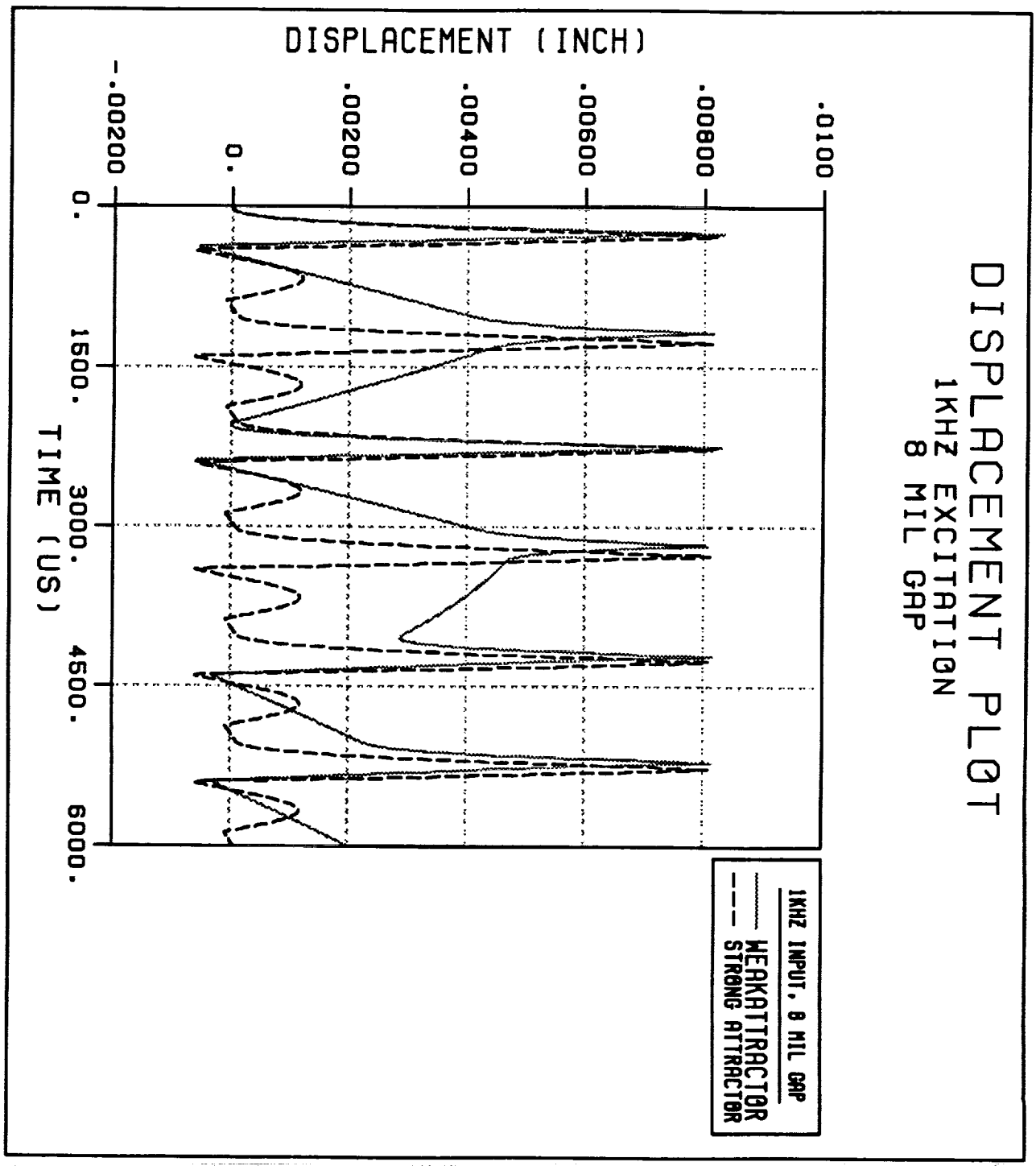


Fig. (26): Phase plots for different linear spring rates

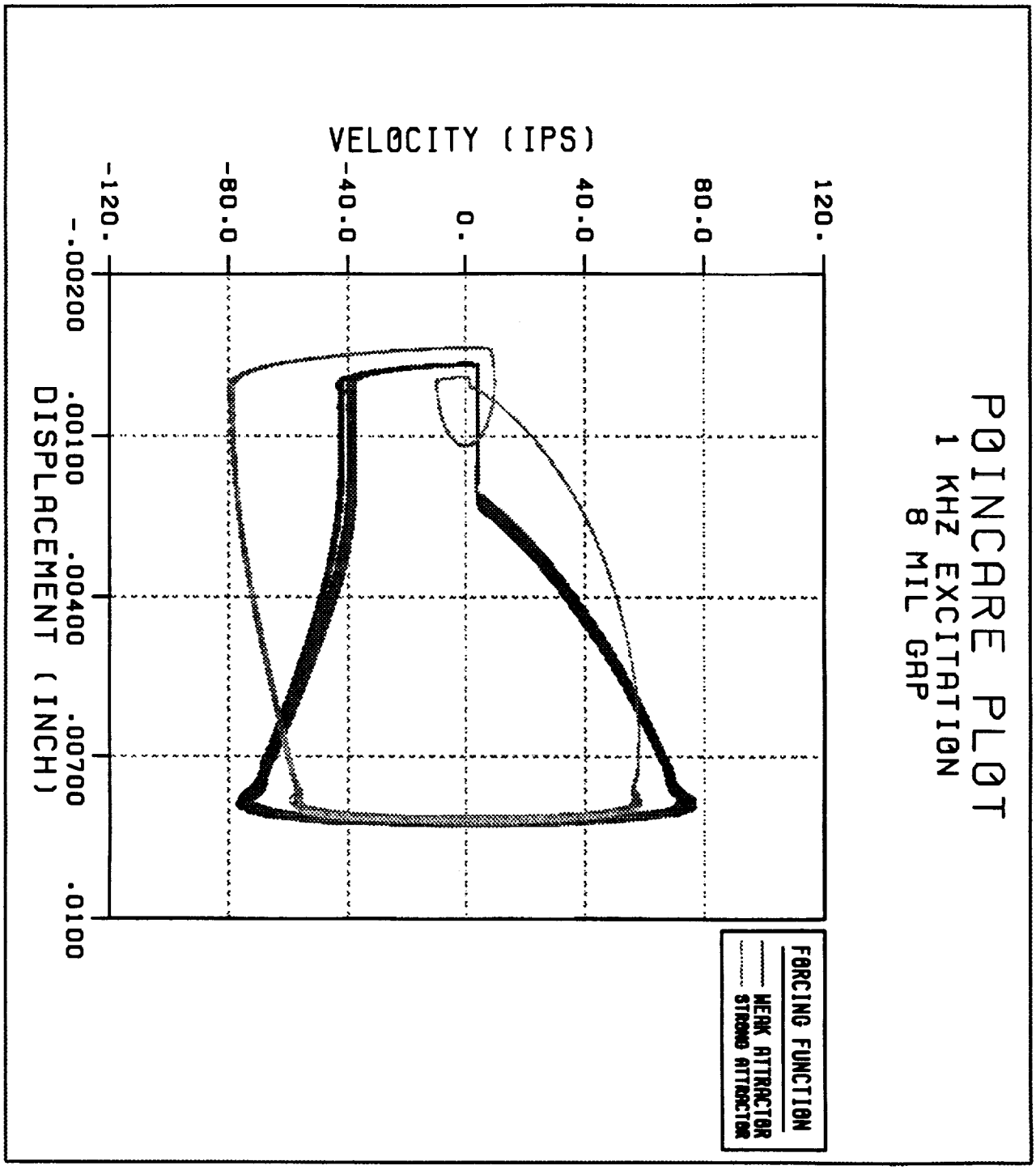
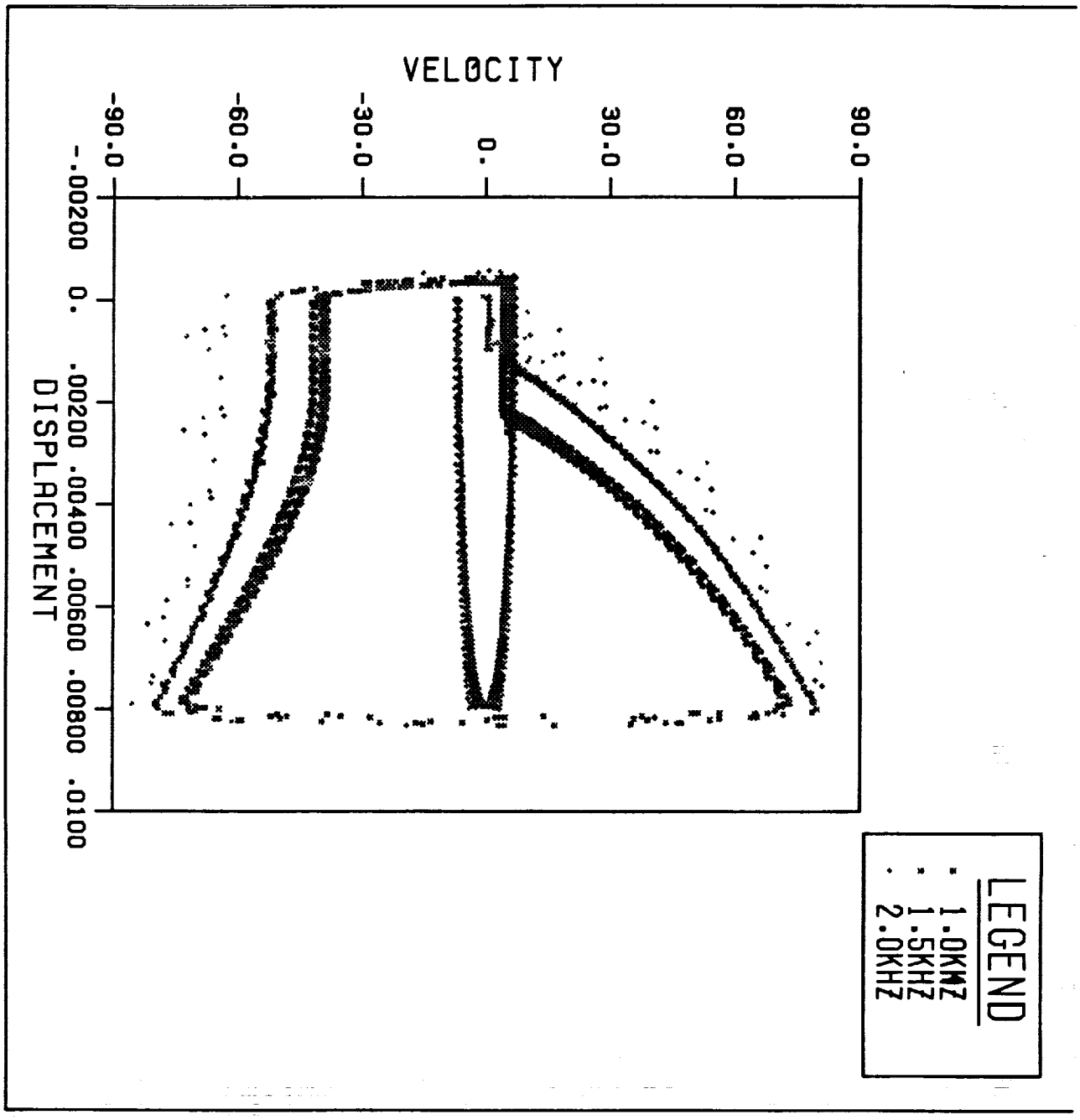


Fig. (27): phase plot at three different forcing freq.s



FEM/SINDA:
Combining the Strengths of NASTRAN, SINDA, I-DEAS, and PATRAN for Thermal and Structural Analysis

N94-17831

P. Richard Zarda, Ted Anderson, Fred Baum

Martin Marietta Missiles Systems
Computer Aided Analysis Group
P.O. Box 5837 MP 135
Orlando, Florida 32855-5837
(407)356-5715

54-39
190574
p. 19

ABSTRACT

This paper describes the interface/integration between FEM/SINDA, a general purpose geometry driven thermal analysis code, and the FEM software: I-DEAS, PATRAN, and NASTRAN. FEM/SINDA brings together the advantages of the finite element method to model arbitrary geometry and anisotropic materials and SINDA's finite difference capability to model thermal properties, loads, and boundary conditions that vary with time or temperature. I-DEAS and PATRAN thermal entities are directly supported since FEM/SINDA uses the nodes of the FEM model as the point at which the temperature is determined. Output from FEM/SINDA (as well as the FEM/SINDA input deck) can be used directly by NASTRAN for structural analysis.

INTRODUCTION

The industry standard thermal analysis codes SINDA and MITAS are known for their versatility in solving a wide range of thermal analysis problems. The input to these codes, however, generally involves tedious hand calculations of nodal capacitances and conductances. The CAE group at Martin Marietta Missile Systems in Orlando, Florida has developed a finite element - finite difference hybrid thermal analysis code which can take finite element models developed in I-DEAS or PATRAN and produce a finite difference network model which is then solved with MITAS, Martin Marietta's version of SINDA (from this point forward, any reference to SINDA implies MITAS as well).

A FEM/SINDA input deck can be generated from an I-DEAS universal file or a PATRAN neutral file using the I-DEAS-to-FEM/SINDA translator or the PATRAN-to-FEM/SINDA translator. FEM/SINDA can then be run to produce *nodal temperatures at the finite element nodes.* The solution algorithm to determine the nodal temperatures is SINDA's finite difference network solution. The conductors and capacitances used in the SINDA network solution are mathematically equivalent to the thermal conductivity and thermal capacitance matrices generated by using finite element techniques. *A node in the finite element model will necessarily be a node in the SINDA model.*

This method will allow, at the I-DEAS or PATRAN level, the mixing of 1-D (rod), 2-D (shell) and 3-D (solid) elements and will generate the conductivity network that this connectivity implies. This is in direct contrast to centroidal methods which require the creation of additional elements when mixing 1-D, 2-D and 3-D elements (for example, the connection between a shell coming into two nodes of a solid requires the creation of one or more shell elements on the face of that solid).

Working with the true finite element nodes (versus the centroidal node) also allows boundary conditions to be easily handled. Specified temperatures can be applied at the finite element nodes which are generated on the true boundary of the object. Applying boundary conditions to centroidal nodes can lead to erroneous answers since the node location is probably not at the proper boundary. In addition, the thermal boundary conditions and loads (such as convection, heat fluxes, radiation, etc.) can be specified in I-DEAS or PATRAN using the current entities available in each of the pre-processors. In I-DEAS or PATRAN the user can also specify whether the properties are isotropic or orthotropic, and whether they are constant or vary with temperature. Boundary conditions and loads are also specified by the modeler to either be constant or vary with time and/or temperature.

FEM/SINDA will automatically generate a SINDA input deck for the subsequent finite difference analysis. This deck can be automatically combined with a SINDA deck that has, for example, a table that could specify how a thermal property (for example, a thermal conductivity already flagged in I-DEAS or PATRAN) would vary with temperature. The complete flexibility of SINDA is therefore available to the thermal analyst. Use of FORTRAN subroutines and tables to account for thermal properties or boundary conditions that vary with time and/or temperature is one of the strengths of SINDA.

FEM/SINDA is also integrated with TRASYS, a well-known code (developed by Martin Marietta) for determining both radiation view factors and solar and planetary heat fluxes. TRASYS has over ten years of development activity and is an industry standard. The I-DEAS or PATRAN user can simply select which faces of shell or solid element radiate. FEM/SINDA will generate the necessary input deck to TRASYS for view factor calculations. A subsequent TRASYS run will return SINDA radiation conductors. These radiation conductors will reflect the view factors between the various radiating elements selected in

I-DEAS or PATRAN. Moreover, the radiation conductors are between the finite element nodes and can be combined with the SINDA deck of thermal conductors for a system analysis involving conduction, convection and radiation.

Existing I-DEAS or PATRAN stress and dynamic models may also be used, with some or no modification, to drive FEM/SINDA. This will then insure, for example, that the temperature field is determined at the nodes of a stress model. A subsequent thermal stress analysis is therefore automatic since nodal temperatures are available. A centroidal method, on the other hand, would require the interpolation/extrapolation of the centroidal temperatures to determine the nodal temperatures – a possible source of misinterpretation and/or error.

Output from FEM/SINDA (either steady state or transient analyses) can be brought back into I-DEAS or PATRAN for processing (also available is the ability to read a FEM/SINDA input deck into I-DEAS or PATRAN). Another feature of FEM/SINDA is that the input deck can be either in free field and/or fixed field, and the card image format is almost identical to a NASTRAN input deck. Existing NASTRAN decks, with slight modification, could therefore be used as input to FEM/SINDA.

In short, the integration of I-DEAS, PATRAN and NASTRAN with FEM/SINDA for thermal analysis combines the power of finite element pre- and post- processing and discretization techniques with the industry accepted SINDA code, taking advantage of the strengths of both while preserving completely the conventional input to SINDA. This allows the FEM user to completely specify his/her thermal model in I-DEAS or PATRAN (conduction, convection, radiation) and allows for boundary conditions, loads and thermal properties to vary with time and/or temperature.

FEM THEORY

In order to understand the basic architecture of FEM/SINDA, a short review of some of the basic techniques in finite element theory is in order. Consider the simple triangular element shown in Figure 1a. The triangle has a constant thickness t and an isotropic thermal conductivity of k. The temperature field within the element is assumed to be a linear function of the nodal temperatures: T_1, T_2, and T_3. It can be shown (see Reference 1) that the temperature field T at any point (x,y) within the element is given by

$$T(x,y) = \frac{1}{2A}\left[a_1 + b_1 x + c_1 y \quad a_2 + b_2 x + c_2 y \quad a_3 + b_3 x + c_3 y\right] \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \end{Bmatrix} \qquad (1)$$

where

$$
\begin{aligned}
A &= \text{ Area of triangle} \\
t &= \text{ Thickness of triangle} \\
a_1 &= x_2 y_3 - x_3 y_2 \\
b_1 &= y_2 - y_3 \\
c_1 &= x_3 - x_2
\end{aligned} \tag{2}
$$

and $a_2, b_2, c_2, a_3, b_3, c_3$, are obtained by permuting the indices in Equation 2 (for example, $b_2 = y_3 - y_1$). If the (x,y) coordinate in Equation 1 equals a nodal coordinate, T(x,y) will reduce to that nodal temperature. Note also that the temperature field of equation (1) is linear.

Next, based on variational principles (Reference 1), the thermal conductivity matrix [K] of the element can be determined. For this triangular element, it is given by (Reference 1)

$$
[K] = \frac{kt}{4A} \begin{bmatrix} (b_1^2 + c_1^2) & (b_1 b_2 + c_1 c_2) & (b_1 b_3 + c_1 c_3) \\ & (b_2^2 + c_2^2) & (b_2 b_3 + c_2 c_3) \\ \text{SYM} & & (b_3^2 + c_3^2) \end{bmatrix} \tag{3}
$$

Note that the matrix is symmetric and not all the values in the matrix are independent. It can be shown (based on the fact that a constant temperature can be maintained with no heat input) that the sum of the values on any row (or column) must add up to zero. Stated another way, the diagonal term on any row is minus the sum of all the off-diagonal terms of that row. Thus, for

$$
[K] = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ & k_{22} & k_{23} \\ \text{SYM} & & k_{33} \end{bmatrix} \tag{4}
$$

once the upper triangular values, k_{12}, k_{13}, and k_{23} are known, all the other entries are determined.

Next consider a conductor network between the same set of nodes as shown in Figure 1b. The conductor values, g_{12}, g_{13}, and g_{23} can be found such that this conductor network is equivalent to the finite element of Figure 1a and Equation 4. This can be shown by recalling that a conductor g between any two nodes A & B has a thermal conductivity matrix given by:

$$[K] = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} \begin{matrix} A & B \end{matrix} \\ \begin{bmatrix} g & -g \\ -g & g \end{bmatrix} \end{matrix} \tag{5}$$

The thermal conductivity matrix for the three conductors of Figure 1b is assembled by applying Equation 5 to each conductor. Then the assembled 3 x 3 conductivity matrix for the three conductors of Figure 1b is

$$[K] = \begin{matrix} \\ 1 \\ 2 \\ 3 \end{matrix} \begin{matrix} \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{bmatrix} G_{12} + G_{13} & -G_{12} & -G_{13} \\ -G_{12} & G_{12} + G_{23} & -G_{23} \\ -G_{13} & -G_{23} & G_{13} + G_{23} \end{bmatrix} \end{matrix} \tag{6}$$

Notice that the conductors are assembled in the matrix consistent with the conduction matrix of Equation 5. Also the matrix exhibits the topology of all conductivity element matrices: the matrix is symmetric and the sum of the off-diagonal terms on any row is equal to minus the diagonal term of that row. Finally, the conductivity matrix of the finite element of Figure 1a will exactly match that of the conductor network of Figure 1b by equating Equation 6 to Equation 4. Only the upper triangular terms need to match (all the others will then necessarily match). This gives

$$\begin{aligned} G_{12} &= -k_{12} \\ G_{13} &= -k_{13} \\ G_{23} &= -k_{23} \end{aligned} \tag{7}$$

Equation 7 simply states that the conductor value between any two nodes i and j is simply minus the off-diagonal i-j term of the thermal conductivity matrix of that element. That is,

$$G_{ij} = -k_{ij} \tag{8}$$

Equation 8 applies not only for the triangular element but for any element. For example, Figure 2a shows a quadrilateral shell element, and the six conductors between the four nodes exactly correspond to the six upper triangle values of the thermal conductivity matrix shown

in Equation 9.

$$[K] = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ & k_{22} & k_{23} & k_{24} \\ & & k_{33} & k_{34} \\ \text{SYM} & & & k_{44} \end{bmatrix} \tag{9}$$

For any element (rod, shell or solid) the thermal conductivity matrix can be determined and the conductivity network is given by Equation 8. The thermal conductivity matrix for most elements can be found in either Reference 1 or Reference 2. Once the conductor network for each element is determined, FEM/SINDA looks towards SINDA (a finite difference code) for solving the system of equations. This is in direct contrast to a finite element code that generally solves a linear system of equations of the form

$$[K]\{T\} = \{Q\} \tag{10}$$

where [K] is a thermal conductivity matrix of size N (N is the total number of nodes in the model), $\{T\}$ is a vector of nodal temperatures, and $\{Q\}$ is a vector of nodal heat flows. The finite element method requires first the assembly of the system thermal conductivity matrix [K] of Equation 10 and then the simultaneous solution to the set of Equations 10. FEM/SINDA does not assemble the matrix [K]. It simply determines the conductivity matrix of an individual element and then generates the appropriate SINDA conductors. The SINDA conductors can vary with time or temperature and hence handle nonlinearities that are common in thermal analysis. On the other hand, finite element techniques are not nearly as efficient (or even capable) in handling nonlinearities (NASTRAN thermal analysis package, for example, is significantly slower than SINDA in solving nonlinear transient problems, and will not handle something as fundamental as a heat transfer coefficient varying with time).

A code such as SINDA requires as input the conductor value between two nodes. For the triangular element of Figure 1a, Equation 3 (applying Equation 8) gives the conductor values. Thus the three conductors are

$$\begin{aligned} G_{12} &= k\left[\frac{t}{4A}(b_1 b_2 + c_1 c_2)\right] \\ G_{13} &= k\left[\frac{t}{4A}(b_1 b_3 + c_1 c_3)\right] \\ G_{23} &= k\left[\frac{t}{4A}(b_2 b_3 + c_2 c_3)\right] \end{aligned} \tag{11}$$

These values (those given by Equation 11) can be input into SINDA in one of two ways. If the thermal conductivity k is constant, FEM/SINDA will generate the following SINDA card:

CONDUCTOR #, NODE$_i$, NODE$_j$, G$_{ij}$
EXAMPLE: 37, 2, 3, 4.278

where the CONDUCTOR # is some unique label number, NODE$_i$ and NODE$_j$ are the nodes that the conductor is between, and G$_{ij}$ is the conductor value which is given by Equation 11. If k is not constant (but is to vary with temperature) the following SINDA card is generated by FEM/SINDA:

CGS CONDUCTOR #, NODE$_i$, NODE$_j$, ARRAY #, (A/L)$_{ij}$
EXAMPLE: CGS 97, 1, 3, A4, 0.789

where CGS implies a conductor that will vary, the ARRAY # (in the example, array A4) is a table of conductivity vs. temperature that specifies how the thermal conductivity is to vary with temperature, and (A/L)$_{ij}$ (a single number) is the "geometric part" of the conductor and is the term in brackets in Equation 11. The table of k vs. T is added separately to the SINDA deck.

Capacitance for each node of each element uses the "lumped mass" approach that is often used in finite element structural analysis. Essentially this means that, for the triangular element of Figure 1a, each node is assigned 1/3 of the mass of that element. For other elements the lumping of mass (and hence capacitance) is similar and can be found in Reference 1 and 2. FEM/SINDA will automatically generate the appropriate capacitance for SINDA.

The above procedure for determining the "finite element" conductors and capacitances for each element is used in a similar way to handle convection and radiation. Convection and radiation will lead to additional conductors in the network and will automatically be generated by FEM/SINDA. SINDA radiation conductors can also include view-factor calculations based on a TRASYS run (the conductors are automatically generated by the TRASYS run).

FEM/SINDA will generate the conductors for each element used in the FEM model. When two elements produce conductors between the same nodes, those conductors are combined (in cases where the conductors are not constant but are referencing a different thermal conductivity, they are not combined). The sorting and summing is performed by FEM/SINDA not only for conductors (conduction, convection and radiation conductors) but also for capacitance and loads. This will generate a compact conductor network for the subsequent thermal analysis.

I-DEAS and PATRAN MODELING

The thermal analyst can define his/her entire thermal model within I-DEAS or PATRAN and then subsequently generate a FEM/SINDA input deck. The key to the ease of generating a FEM/SINDA input deck from a FEM model is simple: a node in the FEM model will necessarily be a node in the SINDA network. The I-DEAS entities available in I-DEAS 4.0 and the PATRAN entities available in PATRAN 2.3 will, in general, be used to directly to drive the FEM/SINDA model. In particular, the I-DEAS and PATRAN entities shown in Table 1 are directly supported by FEM/SINDA.

I-DEAS/PATRAN entity	FEM/SINDA entity
Cartesian coordinate system	CORDR
Cylindrical coordinate system	CORDC
Spherical coordinate system	CORDS
Isotropic material table	MATI
Orthotropic material table	MATO
Spring physical table	PCOND
Rod/Bar physical table	PROD
Shell physical table	PSHELL
Solid physical table	PSOLID
Node	NODE
Node-to-node translational spring	CONDUCT
Lumped mass	CAPAC
Linear rod/bar	ROD
Linear thin-shell triangle	TRIA
Linear thin-shell quadrilateral	QUAD
Linear solid tetrahedron	TETRA
Linear solid wedge	WEDGE
Linear solid brick	BRICK
Nodal heat source	NHEAT
Edge influx/Dist. heat source	EFLUX
Face influx/Dist. heat source	FFLUX
Distributed heat generation	VFLUX
Edge convection	ECNVECT
Face convection	FCNVECT
Edge radiation	ERADS
Face radiation	FRADS or FRADT
Nodal temperature	TEMP

TABLE 1. I-DEAS/PATRAN entity and corresponding FEM/SINDA entity.

The property and material values in I-DEAS can be used and the corresponding FEM/SINDA input deck will be properly generated. Some of the material values that are supported in I-DEAS directly are isotropic and orthotropic thermal conductivity, specific heat and material density. Convective heat transfer coefficients and the emmisivities (for radiation calculations) are also input directly in I-DEAS in the "ANALYSIS_CASES" task. Note that by supporting both edge entities and surface entities (as Table 1 shows) both 2-D and 3-D models can be fully generated in I-DEAS and analyzed by FEM/SINDA. Heat loads, convection and radiation can be applied using I-DEAS's heat transfer loads (see Table 1). I-DEAS's modeling of conductivity, specific heat, loads and boundary conditions that vary with time or temperature is supported by entering a negative integer value for that property. The FEM/SINDA translator (which translates a universal file into a FEM/SINDA input deck) interprets all negative integer values for conductivity, specific heat, loads and boundary conditions as a SINDA array reference (the SINDA array # is the absolute value of the integer). The SINDA input deck must then include an array which describes how that value is to vary with time or temperature.

The PATRAN interface to FEM/SINDA supports almost all FEM/SINDA entities which, like I-DEAS, allows the user to input the entire model in the preprocessor. Nodes and elements are generated with the standard GFEG and CFEG commands. Element properties and material properties are entered with PROP and PMAT commands, respectively. Two PMAT options are supported: thermal isotropic (TIS) and thermal anisotropic (TAN). Material properties which vary with temperature may reference a SINDA array by entering a negative array number in the PMAT command for that property. Boundary conditions are entered with the standard DFEG command and may reference a SINDA time-varying array by entering the array number in the UID field of the DFEG command. The only exception is convection in which the array reference goes in the data field and the convection option (time or temperature dependent) goes in the UID field.

I-DEAS or PATRAN modeling used in conjunction with FEM/SINDA allows the thermal analyst to easily model his/her problem with the tools that are available in the pre-processors. The mapped and free mesh generation, application of loads and boundary conditions to geometric entities, mixing of rod, shell and solid elements are just a few of the FEM's features that can be used (without playing games) to generate a thermal model. Fundamental tasks such as free edge plots can be used meaningfully to show the absence of thermal connections (this is in direct contrast to centroidal methods and any method which does not use the nodes of the finite element model as the point at which the temperature is to be determined). FEM/SINDA's interface with I-DEAS and PATRAN truly allows the modeler to use the pre-processor software consistent with its design, and hence makes the thermal analyst more efficient in his/her modeling.

FEM/SINDA INPUT DECK

The FEM/SINDA translator will read a I-DEAS universal file or a PATRAN neutral file of a thermal model and generate a FEM/SINDA input deck. The input deck to FEM/SINDA looks similar to a NASTRAN input deck (hence present NASTRAN decks can be used, with slight modification, to perform a thermal analysis). Figure 3 shows a quick reference guide describing a FEM/SINDA input deck, and Figure 5 shows an input deck for the simple problem shown in Figure 4. This deck was completely generated from the I-DEAS model shown by first generating a universal file from I-DEAS and then running the FEM/SINDA translator (similiar techniques apply for PATRAN). The card image input is self explanatory (Figure 3 can be used as a quick guide for the field description). The "SFILE" shown in Figure 5 is the name of a supplementary SINDA file. The SFILE can contain SINDA array definitions, FORTRAN subroutines, etc. that will augment the conductor network generated from the finite element model to produce the SINDA input deck. This file could contain old SINDA decks that will be thermally combined with the new finite element input deck. The ECNVECT card shown in Figure 5 defines the heat transfer coefficient to the air gap (see Figure 4) as a function of temperature to be defined by array "A1". This array is specified in the SFILE.

The quick reference guide (Figure 3) indicates which fields of the data input can vary (data enclosed in {}) with time or temperature and hence reference an array. For example, the edge flux card (EFLUX) allows for the flux to be specified by an array.

Radiation is specified (for which TRASYS will calculate the view factors and generate the nodal radiation conductors) by the FRADT card. The radiation conductor network returned from TRASYS is included with the SINDA input deck to form a complete system network which models the integration of the conduction, convection and radiation thermal model.

Once the SINDA analysis is complete, a universal file or neutral file is generated by SINDA that contains all of the nodal temperatures for post-processing. In a transient thermal analysis, this file will contain a temperature data set for each output time step. SINDA can also generate a set of NASTRAN "TEMP" cards that can be included with a NASTRAN input deck for thermal stress analysis.

FEM/SINDA EXAMPLES

The following examples of FEM/SINDA will help to illustrate the advantages of the I-DEAS /PATRAN-to-FEM/SINDA combination to the thermal analyst.

Figure 6 shows the temperature contours for a 2-D model of a rectangular region. Heat is input at the bottom of the region and the top is held at a constant temperature of

zero. The thermal conductivity is constant. The grid shown (using 2-D shell elements) was purposely made irregular to illustrate the strength of the finite element method. The temperature field is linear for this model. FEM/SINDA will model a linear temperature field exactly because of the finite element description of the conductor network (see Equation 1). Three contour plots are shown: (a) FEM/SINDA results, (b) TMG results , and (c) a centroidal method. The FEM/SINDA results give the exact solution, and the TMG results are reasonably close to the exact solution. TMG uses a single "thermal node" per element, but the "node" is not at centroid but at the intersection of the perpendicular bisectors of the sides (assuming a triangular element - a quadrilateral can be broken up into triangles). It can be shown that these "thermal node" points will model a linear temperature field exactly. The apparent discrepancy (from Figure 6) is that TMG will not use these points when the bisector intersection falls outside the triangle. The resulting TMG conductor network is therefore not guaranteed to model the temperature field exactly (a trivial change to the code could remedy this). Despite this, the TMG temperature field is acceptable. This is not the case for the the centroidal temperature field show in Figure 6c. The conductor network for this model is based on an in-house code that uses the centroid as the "temperature node". The results are unacceptable and clearly show that the irregular finite element grid dramatically affects the results (a rectangular grid would give the exact solution).

If the analyst were to use a centroidal method (rather than FEM/SINDA), the modeling for a large model could be complicated and very cumbersome. For example, besides the needed shell elements shown in Figure 6 to model the 2-D conduction region, bar elements must be used at the top and bottom boundary to designate the boundary conditions. This process carried over to 3-D models requires shell elements to be put on the face of solid elements to handle boundary conditions–a process that can add significant modeling time and that is cumbersome. These "additional" elements are sometimes needed even within a solid region; for example, at the interface of two materials with different conductivity. Failure to do so will can cause interpolation algorithms to inadequately predict finite element nodal temperatures from the "element" temperatures. Modeling convection and radiation can also require the addition of elements on the appropriate boundaries. Mixing of 1-D, 2-D, and 3-D conduction elements also requires the "additional" elements when such elements join (a 2-D shell coming into two nodes of a solid requires the addition of a shell on the face of that solid to force the thermal connectivity). Overall, these thermal "games" can significantly affect the thermal analyst's productivity in I-DEAS or PATRAN and can hinder the graphical verification of his model.

Figure 7 shows a radiation-conduction problem that was performed both with FEM/SINDA and NASTRAN. The top body is held at a constant temperature of 100 degrees and the bottom body at 0 degrees. The circular region has a low thermal conductivity and a unit depth is used. Space is at a temperature of 50 degrees. This model was generated in I-DEAS including the designation of the radiating surfaces. FEM/SINDA generated the TRASYS run which produced the view factors and the SINDA radiation conductors. Good agreement

is shown between FEM/SINDA and NASTRAN for the relatively coarse grid used.

Figure 8 shows an example of a transient analysis. It consists of a splice ring used to attach sections of a missile together. Normally, the thermal protection requirements of a missile are determined by a 1-D analysis through a typical portion of the missile skin. In this case two dimensional effects are considered important where the splice section and the bolt area join. For this example, a 2-D mapped mesh model was constructed. Different thermal properties were used for the splice ring, bolt and filler elements. Aerodynamic heating was applied to the outer surface (top) by means of a time-varying adiabatic wall temperature and convection coefficient. The outer surface was also allowed to radiate to the sky. The inner surface (bottom) had constant free convection applied. The results of the five second transient analysis are shown in two forms – four temperature contour plots at various points in time and as a surface temperature versus time plot. The surface temperature time trace compared favorably with the results from a 1-D in-house finite difference code, called F86, which is also shown in the plot.

A practical example showing the use of FEM/SINDA is the model of a TV camera of an electro-optical system that is shown in Figure 9. This model is composed of 2849 nodes and 2834 elements which generated 19289 SINDA conductors (the largest model to date with FEM/SINDA was 4897 nodes and 5423 elements). The model shown is a mixture of rods, shells, and solid elements. Convection loads the exposed surfaces. Heat is input in the mounting bracket (shown in the foreground) because of a direct connection between the bracket and an electronics module. The results shown here represent the steady state temperature distribution. The detail shown in the finite element model was needed for structural analysis. The deflections of the optical train were driven by the temperature distribution. The determination of the temperature distribution at the finite element by FEM/SINDA) made the interface between the structural and thermal model a trivial matter. The other important feature that is automatic in this model was the mixing of various element types. For example, a rod coming into one node of a shell is thermally allowed and easily modeled in PATRAN or I-DEAS. This connectivity is also easily verified in PATRAN or I-DEAS.

CONCLUSIONS

FEM/SINDA provides a general purpose geometry driven thermal analysis code to the thermal analyst. Because of the finite-element-type input to the code (essientially identical to a NASTRAN input deck), its interface with I-DEAS, PATRAN and NASTRAN is complete: each FEM/SINDA entity corresponds naturally to a I-DEAS or PATRAN entity, and in most cases to a NASTRAN entity (hence NASTRAN input decks can, with little or no modification, be used as an input deck to FEM/SINDA). The nodal temperatures determined from FEM/SINDA can be used directly to drive a thermal loading condition in NASTRAN. FEM/SINDA combines the power of finite element techniques with the thermal community's

tested and well accepted workhorse: SINDA. This mix of the finite element-finite difference worlds takes advantage of the strengths of both methods: the finite element method's ability to handle arbitrary geometry, model non-homogeneous regions with different element types, and model linear temperature fields exactly; and SINDA's finite difference capability to handle time and temperature dependent material properities, loads, and boundary conditions, and add user written FORTRAN routines. FEM/SINDA's interface with I-DEAS and PATRAN allows the thermal analyst to take full advantage of all of a finite element modeler's capabilities in a manner consistent with the design of the FEM pre- and post-processors. The key to that interface/integration is that a node of the finite element model will necessarily be a node in the thermal conductor network. Therefore this technique does not comprimise the inherent modeling integrity of FEM geometric discretization, and will easily allow the alogrithms of both old and new finite element technology (for example, both in meshing applications and finite element matrix manipulations) to be applied to general purpose thermal analysis.

REFERENCES

(1) The Finite Element Method, O. C. Zienkiewicz, McGraw-Hill Book Company, 1977

(2) Concepts and Applications of Finite Element Analysis, Robert D. Cook, John Wiley & Sons, Inc., 1981

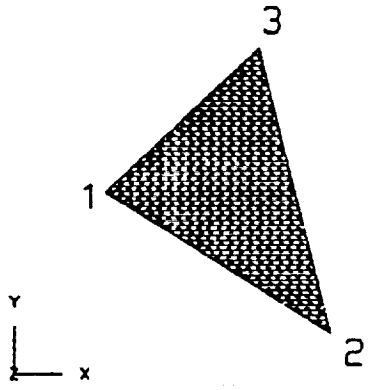


Figure 1a. Triangular Element

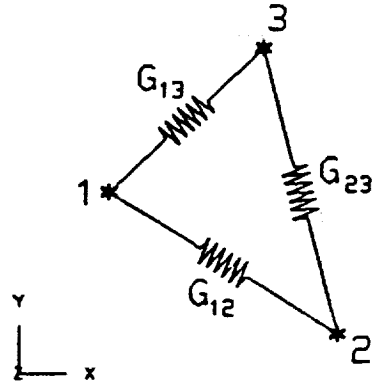


Figure 1b. Equivalent Conductors

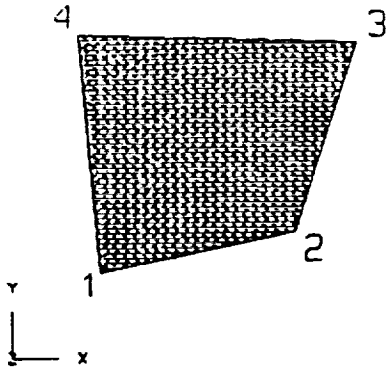


Figure 2a. Quadrilateral Element

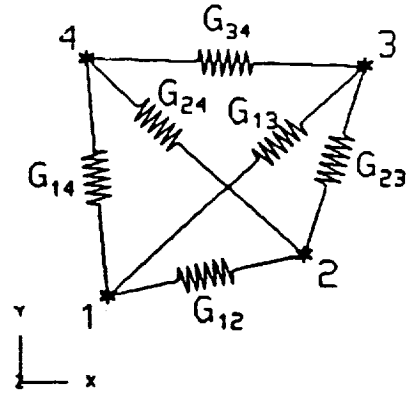


Figure 2b. Equivalent Conductors

```
TITLE=        $ Give some title
OUTPUT=       $ SUPERTAB, PATRAN, NASTRAN, and/or PLOT
ITEMP=        $ Initial temperature
RTEMP=        $ Temperature of radiation infinity node
SFILE=        $ Name of Sinda_formatted input file
MINCOND=      $ Minimum abs(A/L) value to be written
BEGIN BULK
$
$ Data enclosed in [ ] may be either a real constant
$ or reference an array
$
$ Coordinate systems:
CORDR, CID, A1, A2, A2,, B1, B2, B3, +Ccid
+Ccid, C1 , C2, C3
CORDC, CID, A1, A2, A2,, B1, B2, B3, +Ccid
+Ccid, C1 , C2, C3
CORDS, CID, A1, A2, A2,, B1, B2, B3, +Ccid
+Ccid, C1 , C2, C3
$
$ Nodes:
NODE    ID      CID     X       Y       Z
FNODE   ID
$
$ Material properties:
MATI    MID     [K]     rho     [Cp]
MATO    MID     [Kx]    [Ky]    [Kz]    rho     [Cp]
$
$ Physical properties:
PCOND   PID     [KA/L]
PROD    PID     MID     AREA
PSHELL  PID     MID     THKNESS
PSOLID  PID     MID
$
$ Elements:
CAPAC   N1      MASS    [Cp]
CONDUCT N1      N2      [KA/L]
FCOND   FN1     FN2     [MDOT]  [Cp]
$
ROD     EID     PID     N1      N2
TRIA    EID     PID     N1      N2      N3
QUAD    EID     PID     N1      N2      N3      N4
TETRA   EID     PID     N1      N2      N3      N4
WEDGE, EID, PID, N1, N2, N3, N4, N5, N6
BRICK, EID, PID, N1, N2, N3, N4, N5, N6, +Beid
+Beid, N7, N8
$
$ Boundary Conditions:
NHEAT   NODE    [HEAT]
EFLUX   EID     EDGE#   [FLUX1] FLUX2
FFLUX   EID     FACE#   [FLUX1]
VFLUX   EID     [FLUX1]
ECONVECT EID    EDGE#   [h1]    h2      FNODE   TYPE
FCONVECT EID    FACE#   [h]     FNODE   TYPE
ERADS   SID     EID     EDGE#
FRADS   SID     EID     FACE#
ENCLOSE SID1    e1      SID2    e2      F12     TB
FRADT   SID     EID     FACE#   e       I
TEMP    NODE    [TEMP]
```

Figure 3. General Description of FEM/SINDA Input Deck

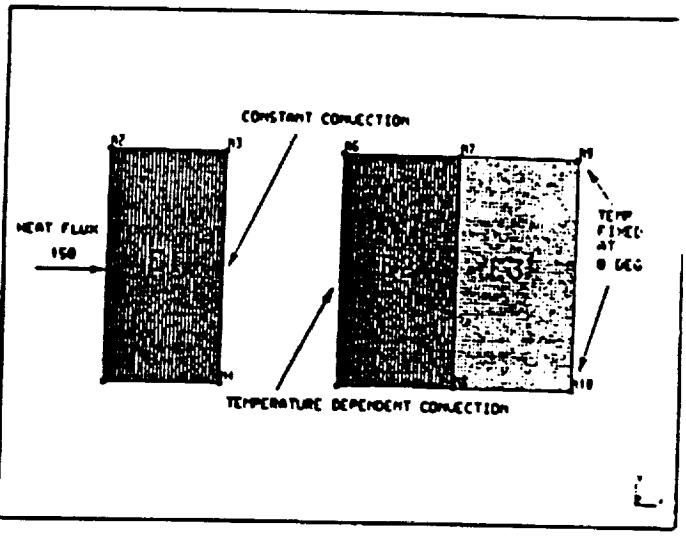


Figure 4. Simple Model Generated in SUPERTAB

```
TITLE = PLATE MODEL SAMPLE CASE
ITEMP = 50.0                    $ Initial temperature
SFILE = PLATE_SAMPLE.MIT        $ Sinda-formatted input file
OUTPUT = SUPERTAB               $ SUPERTAB Unv. file output
MINCOND= 1.0e-15                $ Eliminate all conductors
$                                 with ABS(A/L) < 1.0e-15
BEGIN BULK
MATI*           1      10.     200.      .2
MATI*           2      20.     100.      .4
PSHELL          1       1      0.5
PSHELL          2       2      0.5
NODE            1       0      0.0       0.0       0.0
NODE            2       0      0.0       4.0       0.0
NODE            3       0      2.0       4.0       0.0
NODE            4       0      2.0       0.0       0.0
NODE            5       0      4.0       0.0       0.0
NODE            6       0      4.0       4.0       0.0
NODE            7       0      6.0       4.0       0.0
NODE            8       0      6.0       0.0       0.0
NODE            9       0      8.0       4.0       0.0
NODE           10       0      8.0       0.0       0.0
QUAD            1       1        1         2         3         4
QUAD            2       1        5         6         7         8
QUAD            3       2        8         7         9        10
TEMP            9     0.0
TEMP           10     0.0
EFLUX           1       1     150.      150.
ECONVECT        1       3      10.       10.      1000         0
ECONVECT        2       1      A1                 1000         1
```

Figure 5. Input Deck Generated from Model Shown in Figure 4

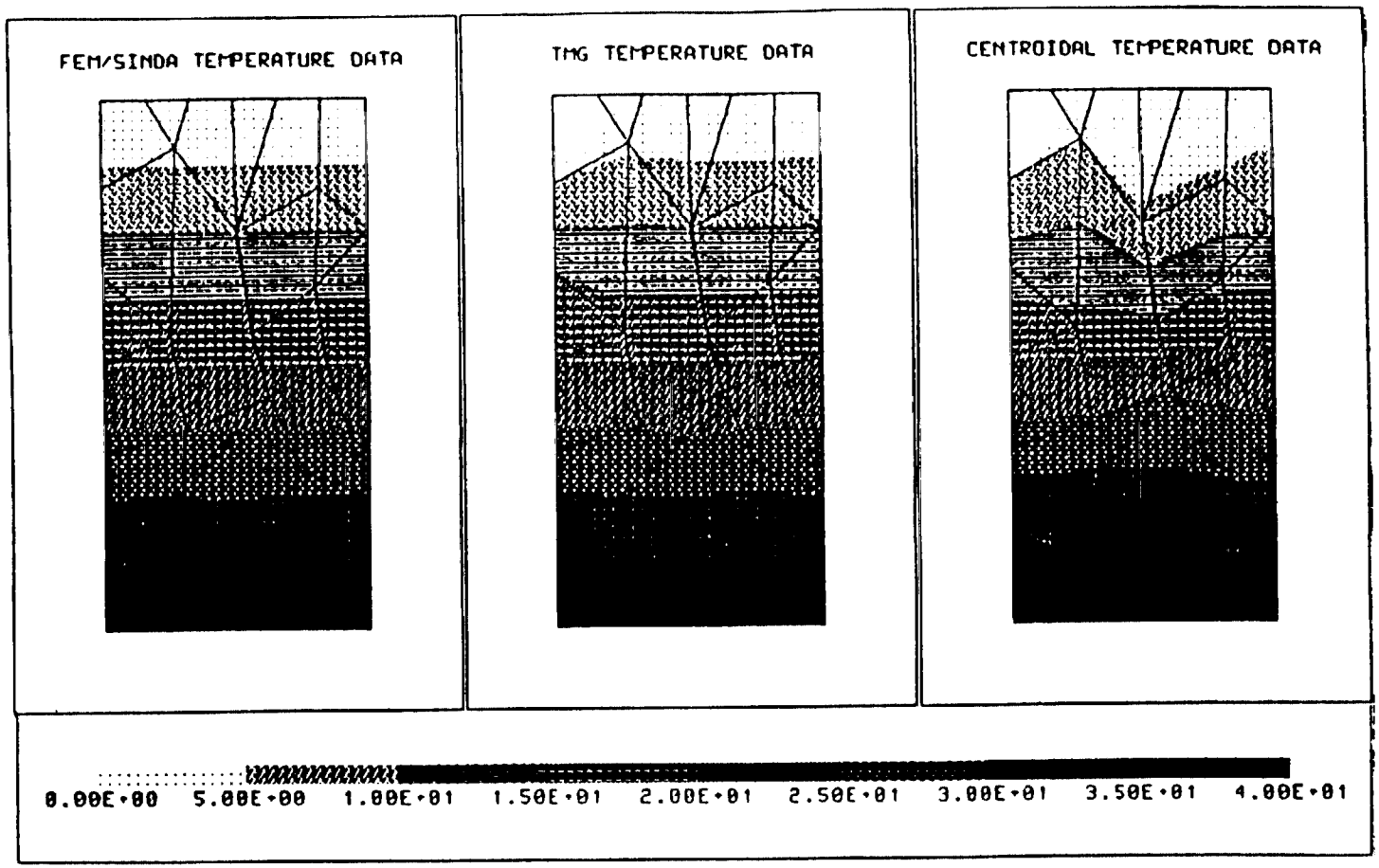


Figure 6. Linear Temperature Field Example

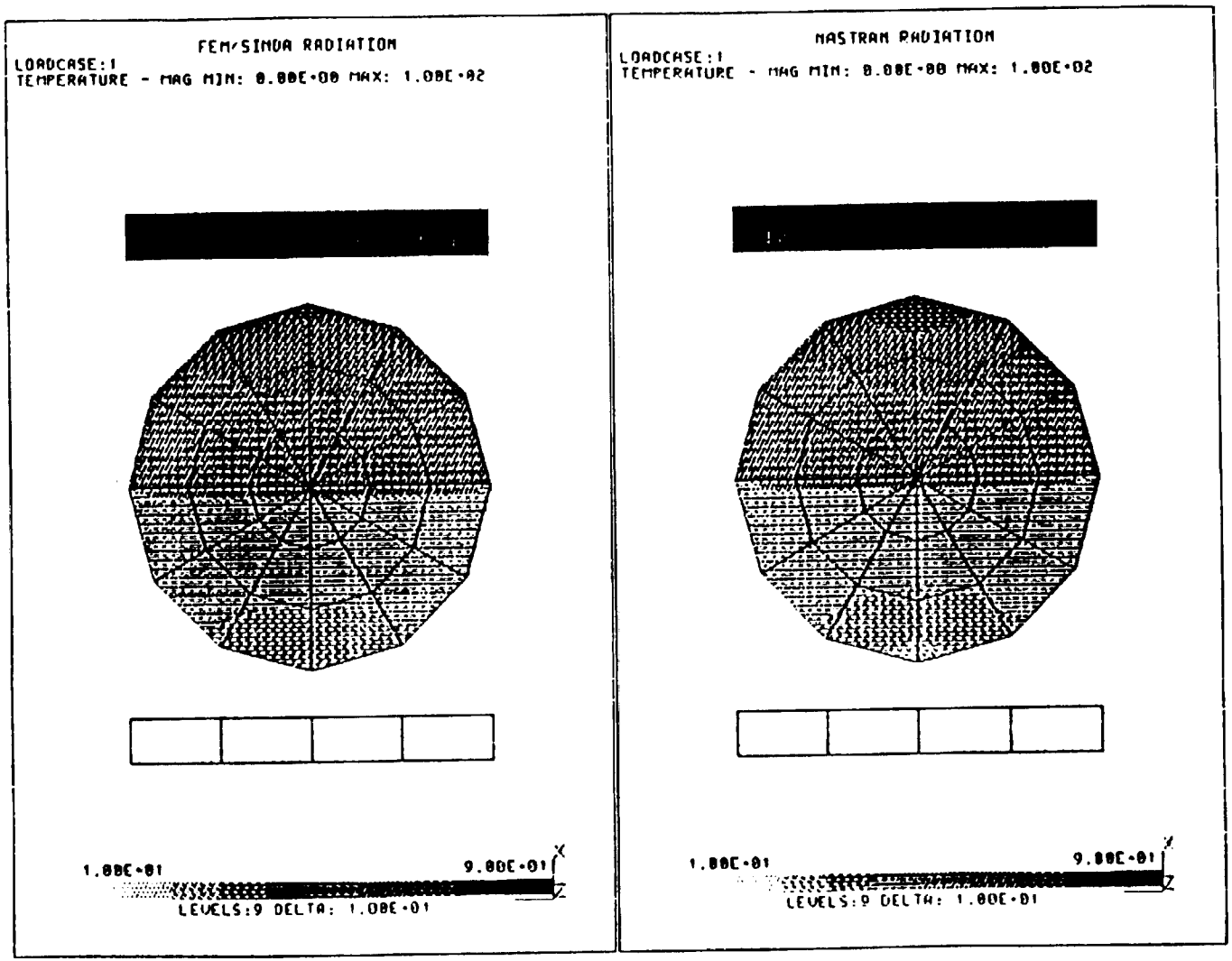
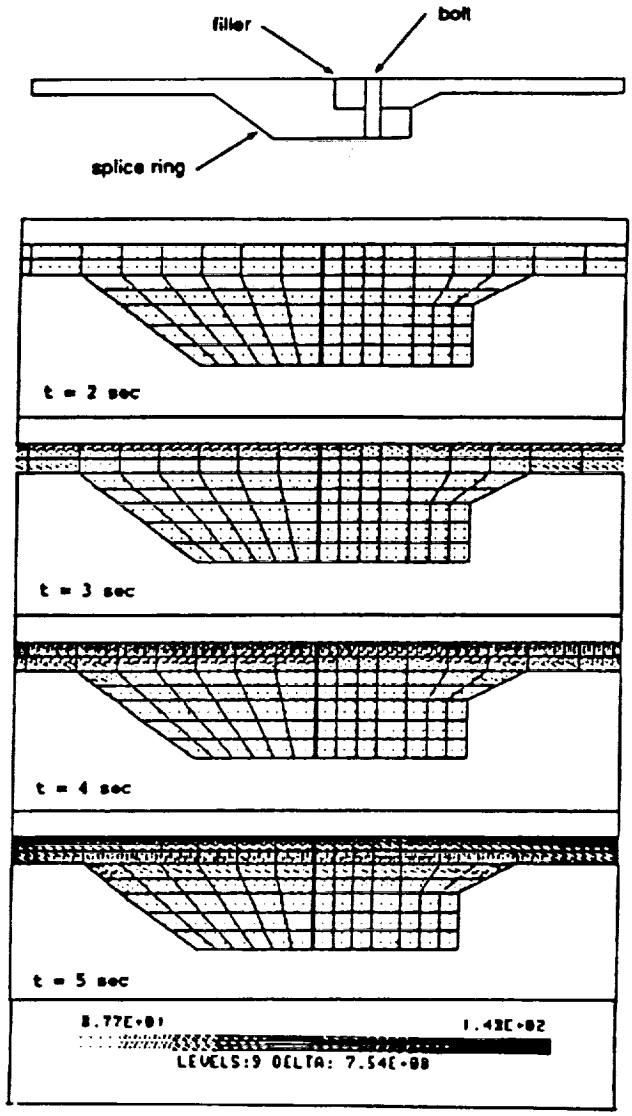


Figure 7. Radiation-Conduction Example



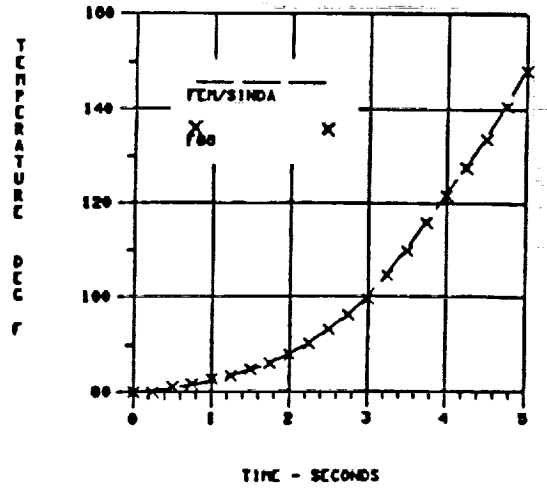


Figure 8. Thermal Transient Analysis of a Missile Splice Ring

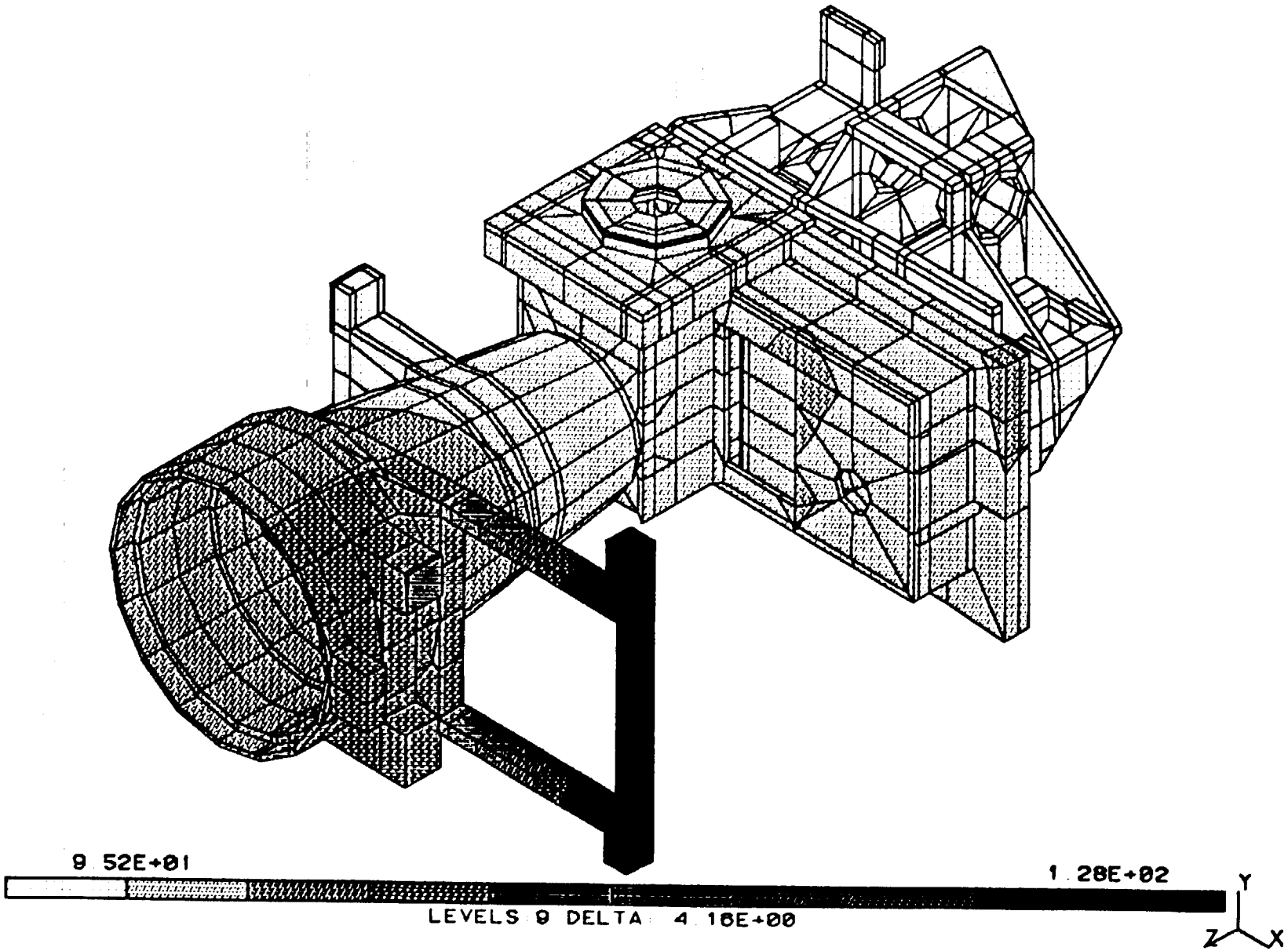


Figure 9. TV Camera Model

GENERALIZED SEISMIC ANALYSIS

Thomas G. Butler
Butler Analyses

INTRODUCTION

There is a constant need to be able to solve for enforced motion of structures. Spacecraft need to be qualified for acceleration inputs. Truck cargoes need to be safequarded from road mishaps. Office buildings need to withstand earthquake shocks. Marine machinery needs to be able to withstand hull shocks. All of these kinds of enforced motions are being grouped together under the heading of seismic inputs.

Attempts have been made to cope with this problem over the years and they usually have ended up with some limiting or compromise conditions. The crudest approach was to limit the problem to acceleration occurring only at a base of a structure, constrained to be rigid. The analyst would assign arbitrarily outsized masses to base points. He would then calculate the magnitude of force to apply to the base mass (or masses) in order to produce the specified acceleration. He would of necessity have to sacrifice the determination of stresses in the vicinity of the base, because of the artificial nature of the input forces.

The author followed the lead of John M. Biggs[1] by using relative coordinates for a rigid base in a 1975 paper[2], and

---

1. "Introduction to Structural Dynamics" by John M. Biggs, McGraw Hill 1964, Sec 6.2.
2. "Fidelity in Shaker Simulation Analysis with NASTRAN", T. G. Butler, January 1975, Orally presented at the First MSC NASTRAN Colloquium.

again in a 1981 paper[3]. This method of relative coordinates was extended and made operational as DMAP ALTER packets to rigid formats 9, 10, 11, & 12 under contract N60921-82-C-0128. This method was presented at the twelfth NASTRAN Colloquium.[4] Another analyst in the field, Gary L. Fox, develped a method [5] that computed the forces from enforced motion then applied them as a forcing to the remaining unknowns after the knowns were partitioned off. The method was translated into DMAP ALTER's, but was never made operational. All of this activity jelled into the current effort. Much thought was invested in working out ways to unshakle the analysis of enforced motions from the limitations that persisted. In the following theoretical development the avenue to complete generality is charted. The method is in the process of being coded and will be implemented as four new rigid formats.

THEORY

Seismic analysis in the displacement method becomes especially challenging, because forces are required in NASTRAN to provide loading for the dynamic solutions. The attempt here is to admit displacement histories as acceptable loadings by converting them into equivalent force loadings. The development of this theory will start with a statement of the general dynamic equation based upon all freedoms being present before any constraints or reductions are applied; this is known as the P-set

---

3. "Dynamic Structural Responses to Base Acceleration", Thomas G. Butler, Proceedings of the Conference on Finite Element Method & Technology, March 1981; Paper No. 8.
4. "Seismic Analysis Capability in NASTRAN", Thomas G. Butler and Robert F. Strang; Proceedings of the 12th NASTRAN Colloquium, May 7-11, 1984, pp 92 - 131.
5. "Solution of Enforced Boundary Motion in Direct Transient and Harmonic Problems", Gary L. Fox, Proceedings of the Ninth NASTRAN Users Colloquium, Oct 22-23, 1080, pp 96 - 105.

(set of all freedoms obtained from all points, grid and extra) in NASTRAN.

$$\left[\left[M_{PP}\right]p^2 + \left[B_{PP}\right]p + \left[K_{PP}\right]\right]\left\{u_P(t)\right\} = \left\{P_P(t)\right\}, \qquad (1)$$

where lower case p stands for the differential operator d/dt. Freedoms which are directly exposed to seismic forcings (accelerations, velocities, & displacements) will be given the designation "C" (standing for contact freedoms) and the complement of this set with respect to the P-set will be designated "J". The P-set of Equation (1) will be partitioned between J & C to get

$$\left[\left[\begin{array}{c|c} M_{CC} & M_{CJ} \\ \hline M_{JC} & M_{JJ} \end{array}\right]p^2 + \left[\begin{array}{c|c} B_{CC} & B_{CJ} \\ \hline B_{JC} & B_{CC} \end{array}\right]p + \left[\begin{array}{c|c} K_{CC} & K_{CJ} \\ \hline K_{JC} & K_{JJ} \end{array}\right]\right]\left\{\begin{array}{c} u_C(t) \\ \hline u_J(t) \end{array}\right\} = \left\{\begin{array}{c} P_C(t) \\ \hline P_J(t) \end{array}\right\}. \qquad (2)$$

Points will be allowed to be loaded with both displacement and force histories. This will provide for such cases as a space craft being tested in a centrifuge with a shaker on board. In such a case there will be body forces being applied by the centrifuge on all points including contact points, $P_C(t)$, and complement points, $P_J(t)$; and displacement histories being applied by the shaker, $u_C(t)$. Single point constraints (SPC's) can be applied only to J dof's, but multipoint constraints (MPC's) can exist between C & J dof's, however the C freedoms must be chosen as independent when defining the constraint. Thus the known quantities in equation (2) are the forces on the complement set P_J, the forces on the contact set P_C, and the displacement histories at the contact set u_C, pu_C, and p^2u_C.

Since the set of u_C are known, the terms involving them can be expanded from equation (2). Take the known terms in the upper partition first:

$$\left[\left[M_{CC}\right]p^2 + \left[B_{CC}\right]p + \left[K_{CC}\right]\right]\left\{u_C\right\} . \qquad (3)$$

The dimension of each of these 3 terms is force. Designate the set of terms in expression (3) as P_C^C; i.e. the forces from displacement histories on the contact freedoms. Next the known terms in the lower partition expands into:

$$\left[\left[M_{JC}\right]p^2 + \left[B_{JC}\right]p + \left[K_{JC}\right]\right]\left\{u_C\right\} \qquad (4)$$

Designate the set of terms in expression (4) as P_J^C; i.e. the forces on those complement freedoms, J, from displacement histories due to their being coupled to the contact freedoms, C. The first term of expression (3) $\left[M_{CC}\right]p^2\left\{u_C\right\}$ constitutes forces that develop from the accelerations of masses at the contact surface. The first term of expression (4) $\left[M_{JC}\right]p^2\left\{u_C\right\}$ constitutes forces tnat develop in the "complement" set from the accelerations of interior masses due to their couplings with the contact set. The second term of expression (3) $\left[B_{CC}\right]p\left\{u_C\right\}$ constitutes forces from the speeding of dampers that are connected between members of the contact set. The second term of expression (4) $\left[B_{JC}\right]p\left\{u_C\right\}$ constitutes forces that develop in the "complement" set from the speeding of dampers that are connected between the interior and the contact set. The third term of expression (3) $\left[K_{CC}\right]\left\{u_C\right\}$ constitutes forces that develop from the deformation of elastic elements that are connected between members of the contact set. The third term of expression (4) $\left[K_{JC}\right]\left\{u_C\right\}$ constitutes forces that develop in the "complement" set from the deformation of elastic elements that are connected between the interior and contact set. The portrayal of the forces on the interior dof's must be extracted from the J partitioning of the P-set, otherwise an incorrect distribution would result from the increased coupling if they were extracted from a reauced order such as N-set or A-set.

The scheme here is to treat the excitation histories as known only for the purpose of computing forces that develop from displacements on contact points. Once the forces from displacement histories are defined they will be added to boundary force histories to give an array of excitations expressed entirely of forces in spite of the fact that part develop from displacement histories. After the forces from displacement histories are fully defined, the contact freedoms $u_C(t)$ will henceforth be treated as unknown. In effect the scheme is to re-solve for displacement histories that are already known. This can be characterized with the following example. Put simply; if one were to look at a single dof system dynamic equation

$$mp^2x(t) + bpx(t) + kx(t) = P(t) \qquad (5)$$

one could compute the value of the external forcing P(t) if all three of the displacement histories were known. For the opposite case, one could treat P(t) as known in equation (5), and integrate it to find the acceleration, velocity and displacement at any time. The result would be to recover the values that were originally known (assuming perfect differentiation and integration routines). This is not an unreasonable approach in view of the power in today's computers.

With the displacements on contact points being treated as unknowns, the forces in equation (2) can now be augmented with the forces from displacement histories as follows:

$$\left[\left[\begin{array}{c|c} M_{CC} & M_{CJ} \\ \hline M_{JC} & M_{JJ} \end{array}\right]p^2 + \left[\begin{array}{c|c} B_{CC} & B_{CJ} \\ \hline B_{JC} & B_{CC} \end{array}\right]p + \left[\begin{array}{c|c} K_{CC} & K_{CJ} \\ \hline K_{JC} & K_{JJ} \end{array}\right]\right]\left\{\begin{array}{c} u_C(t) \\ \hline u_J(t) \end{array}\right\} = \left\{\begin{array}{c} P_C(t) + P_C^C(t) \\ \hline P_J(t) + P_J^C(t) \end{array}\right\} \qquad (6)$$

$u_C(t)$ would be recovered if $P_C(t)$ & $P_J(t)$ were null.

This lays the groundwork for implementation. Provision must be made for admitting displacement history specifications as bulk data; i.e. $p^2\{u(t)\}$, $p\{u(t)\}$, and u(t). Next, the computation of $P_C^C(t)$ and $P_J^C(t)$ must be provided for. Different parts of a structure can have certain portions involved in a given displacment excitation while other portions could be subject to distinctly different excitations. Thus a framework is needed for the spatial specification of each distinct excitation. There can also be spatially distinct time delays associated with individual excitations. But a mechanism already exists in NASTRAN for such specifications: i.e. DAREA for spatial specification of magnitudes, TABLEDi for time varying amplifications, and DELAY for spatial specifications of time delays. All of these can be used with impunity and without confusion with respect to the normal input of dynamic data by requring unique set ID numbers and by having a seismic assembler of enforced loadings. A new case control command called SEISLOAD and a new bulk data card called SEISLOAD will be put into service. Bulk SEISLOAD will act much like TLOADi and RLOAD cards in organizing the spatial, temporal, and phase aspects of displacement excitations. It will incorporate one additional BCD field to specify the type of displacement being input; DISP, or VELO, or ACCE. SEISLOAD case control command will activate the bulk SEISLOAD card much like the DLOAD case contol command that activates the bulk DLOAD card. The Input File Processor (IFP) will assemble the seismic bulk data into the initial data block called DYNAMICS. Case control will direct the data from its SEISLOAD card to read the data from the DYNAMICS data block with a new functional module SPD (seismic pool distributor) whose function would be similar to the DPD (dynamics pool distributor) to prepare SEISLT (seismic load table) and SEISRL (seismic response list) similar to the DLT & TRL. Now comes the actual work of processing these tables and

lists into actual force histories. SEISLT & SEISRL would be input to a second new module SEISLG (seismic load generator) that would treat each distinct displacement excitation as an individual case. That is, SEISLG would form the partitioning vector of the P-set between the C & J sets for one distinct loading. It would compute the equivalent set of three force loadings and ready it for combining with loads from Load generator modules; then turn to the next distinct case and build another partitoning vector for this succeeding case and proceed as before in computing the equivalent set of three loadings. A record should probably be kept for purposes of checking and in setting up output sets for recovery of proof of re-solving for the input specifications.

There are several situations that must be anticipated. First an important premise must be stated. REGARDLESS OF WHAT COMPONENTS OF SEISMIC EXCIATION ARE SPECIFIED (p^2U, pU, OR U), ALL THREE COMPONENTS EXIST AS A CONSEQUENCE OF THE EXISTENCE OF ANY ONE OF THEM. For example, if a seismic acceleration were given as a specification for excitation, the associated velocity and displacment histories can be derived by integration. All 3 components of a seismic disturbance can produce excitation in a structure provided that the structure contains appropriate elements that are coupled to the contact points. Therefore if only one or two out of the three components are specified, the analysis must be equipped to derive the missing component(s). This means that seismic specifications must be differentiated and/or integrated to complete the description of the excitation. Modules will need to be written to perform both integration and differentiation of these displacement histories. The options would be these when all three components are needed:

(a) Only DISP is specified on the SEISLOAD card.
Consequence: Differentiate twice to obtain seismic velocity and seismic acceleration.

(b) Only VELO is specified on the SEISLOAD card.
Consequence: Differentiate once to get seismic acceleration. Integrate once to get seismic displacement.

(c) Only ACCE is specified on the SEISLOAD card.
Consequence: Integrate twice to get seismic velocity and seismic displacement.

Once the three components of seismic excitation are fully enunciated for one case they will be ready for delivery to SEISLG for computation of forces. Each such triplet of histories must be identified with its associated spatial companion. Some connection must be made with Case Control so as to keep these various combinations of load separated for purposes of managing the solution and data recovery operations.

SEISLG must operate similar to TRLG in that it should produce P-set forces, and D-set forces, and S-set forces. It will do this for the C-set based on the SEISLOAD data. It will also have to determine which of the J-set are loaded and to what extent, due to their individual coupling and prepare these additional loadings. After the dynamic load generator has done its work on normal forcing, the forces due to displacements should be added into the three different partitions of load vectors such as the P_P vector.

$$\left\{P_P\right\} = \left\{P_P + P_P^C\right\} = \left\{\frac{P_C}{P_J}\right\} + \sum_{i=1}^{k} \left\{\frac{P_C^C}{P_J^C}\right\}_i , \qquad (7)$$

where i represents a distinct contact set.

For each C dof there exists a distinct set of coupling to the J dof's for mass and for elasticity, and for damping. Therefore, for each C dof for each C point there will be a distinct C-J partitioning vector. For example, if there are 2 C-points and if each point were being excited in 2 translational dof's, there are 4 possible couplings for mass, 4 possible couplings for damping, and 4 possible couplings for stiffness. Thus there would be 3 x 4 = 12 distinct J-C vectors, 12 distinct DAREA patterns, 12 distinct TLOAD1 combinations, 2 x 2 X 3 = 12 distinct TABLED1 histories, 3 x 4 = 12 DELAY spatial distributions, and 1 SEISLOAD assemblage.

Translated into a specific example, if the two C-points were numbered 50 and 60 and the excitations were in axial (x=1) and transverse (y=2) directions, there will be 4 distinct acceleration histories: 50(x) and 50(y) plus 60(x) and 60(y). The mass coupling between 50(x) and its J neighbors would probably have a different pattern than that of the mass coupling between 50(y), 60(x) and 60(y) and their respective J neigbors. So the DAREA content for the spatial loading from the acceleration excitation at 50(x) will have to be derived from the mass coupling to 50(x). Fortunately the DELAY content for the spatial time lapse of the acceleration history at 50(x) will be the same as the DAREA content for 50(x). Similarly, the DAREA & DELAY distributions for 50(y), 60(x), and 60(y) will have to be derived from the mass couplings between their J neighbors and at the respective points 50(y), 60(x), and 60(y).

This same pattern of reasoning applies to the formation of loadings for displacement histories stemming from stiffness coupling between the C dof's and their J neighbors. And again this same reasoning applies to the formation of loadings for the

velocity histories stemming from damping coupling from the C dof's and their J neighbors. TLOAD1's and SEISLOAD for the 12 loadings can be described thusly:

ACCE @ 50(x) TLOAD1

1	DAREA from mass coupling to 50(x)	TABLED1 from acce history at 50(x)	DELAY from mass coupling to 50(x)

VELO @ 50(x) TLOAD1

2	DAREA from damp coupling to 50(x)	TABLED1 from velo history at 50(x)	DELAY from damp coupling to 50(x)

DISP @ 50(x) TLOAD1

3	DAREA from stiff coupling to 50(x)	TABLED1 from disp history at 50(x)	DELAY from stiff coupling to 50(x)

ACCE @ 50(y) TLOAD1

4	DAREA from mass coupling to 50(y)	TABLED1 from acce history at 50(y)	DELAY from mass coupling to 50(y)

VELO @ 50(y) TLOAD1

5	DAREA from damp coupling to 50(y)	TABLED1 from velo history at 50(y)	DELAY from damp coupling to 50(y)

DISP @ 50(y) TLOAD1

6	DAREA from stiff coupling to 50(y)	TABLED1 from disp history at 50(y)	DELAY from stiff coupling to 50(y)

ACCE @ 60(x) TLOAD1

7	DAREA from mass coupling to 60(x)	TABLED1 from acce history at 60(x)	DELAY from mass coupling to 60(x)

VELO @ 60(x) TLOAD1

8	DAREA from damp coupling to 60(x)	TABLED1 from velo history at 60(x)	DELAY from damp coupling to 60(x)

DISP @ 60(x) TLOAD1

9	DAREA from stiff coupling to 60(x)	TABLED1 from disp history at 60(x)	DELAY from stiff coupling to 60(x)

ACCE @ 60(y) TLOAD1

10	DAREA from mass coupling to 60(y)	TABLED1 from acce history at 60(y)	DELAY from mass coupling to 60(y)

VELO @ 60(y) TLOAD1

11	DAREA from damp coupling to 60(y)	TABLED1 from velo history at 60(y)	DELAY from damp coupling to 60(y)

DISP @ 60(y) TLOAD1

12	DAREA from stiff coupling to 60(y)	TABLED1 from disp history at 60(y)	DELAY from stiff coupling to 60(y)

COMBINED SEISLOAD

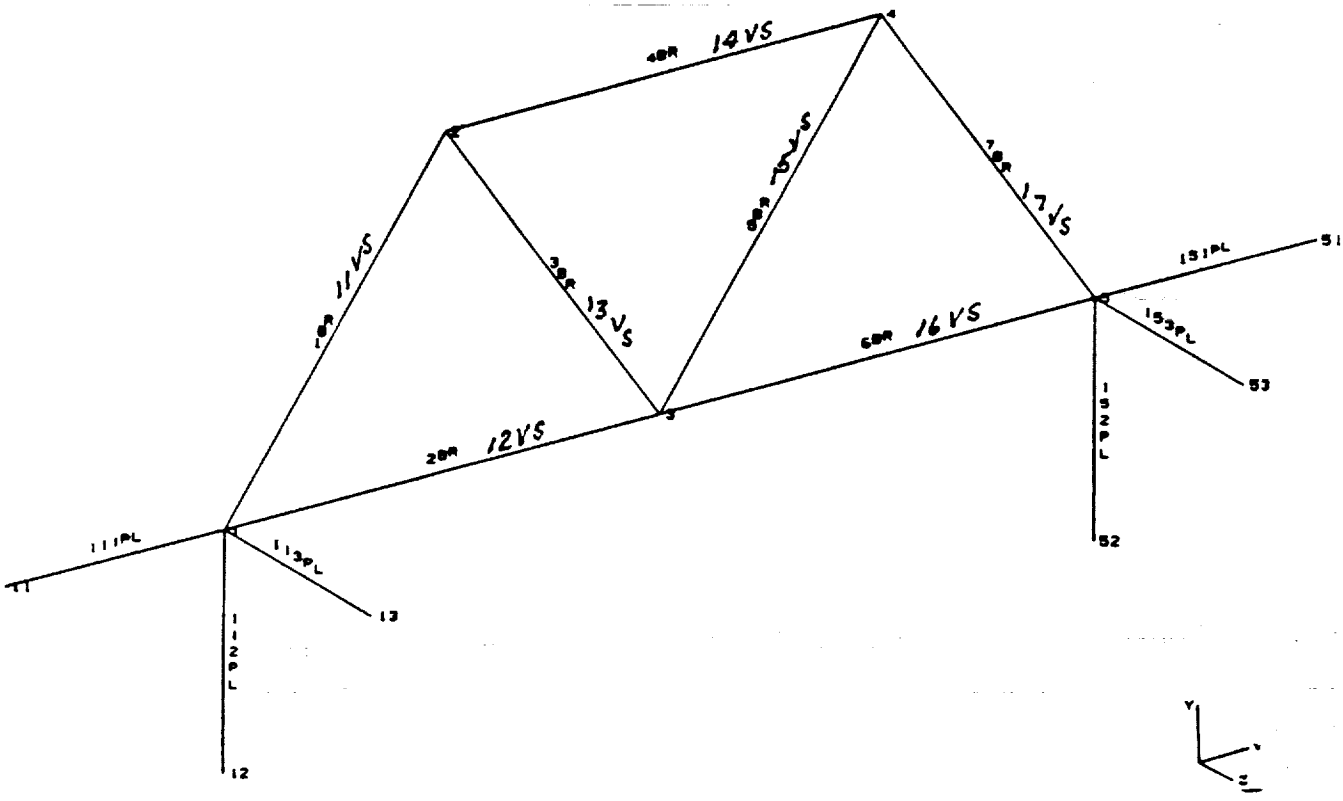
13	1.0	1.0	ACC @ 50(X)	1.0	VEL @ 50(X)	1.0 DIS @ 50(X)
		1.0	ACC @ 50(Y)	1.0	VEL @ 50(Y)	1.0 DIS @ 50(Y)
		1.0	ACC @ 60(X)	1.0	VEL @ 60(X)	1.0 DIS @ 60(X)
		1.0	ACC @ 60(Y)	1.0	VEL @ 60(Y)	1.0 DIS @ 60(Y)

Now all bookkeeping is in the hands of Case Control and the loads are all in terms of force, so the dynamic solution can proceed as it normally does, including the recovery of data. The output should provide bookkeeping for the several C sets that were fed to the SPD (Seismic Pool Distributor module) so that a separate reporting of tnese dynamic displacements can be assembled for comparison with the specified seismic histories anc/or a differencing should take place to give a measure of the effectiveness in re-solving for the specified seismic inputs.

APPLICATION

This theory has been implemented in DMAP form for Direct Transients. Although the problems were small pilot examples they included extra points and DMIG matrices and involved excitations from mass coupling, damping coupling and stiffness coupling. The theory has been thoroughly certified. The pilot problem, shown

in the plot, represents a simple truss bridge on three foundations with a seismic wave travelling in the positive x direction and disrupting these foundataions.



CONCLUSION

Here at last is an automatic method for handling enforced motion that is completely general. The method has been shown to be operational in a DMAP mode. There is no special burden on the analyst except to provide the usual engineering information giving the particulars of his problem. The coding will be completed by the summer of 1993 and will be available in the 1994 release of NASTRAN.

56-39

190576

A Noniterative Improvement Of Guyan Reduction

N. Ganesan

GE Government Services, Houston, Texas

ABSTRACT: *In determining the natural modes and frequencies of a linear elastic structure, Guyan reduction is often used to reduce the size of the mass and stiffness matrices and the solution of the reduced system is obtained first. The reduced system modes are then expanded to the size of the original system by using a static transformation linking the retained degrees of freedom to the omitted degrees of freedom. In the present paper, the transformation matrix of Guyan reduction is modified to include additional terms from a series accounting for the inertial effects. However, the inertial terms are dependent on the unknown frequencies. A practical approximation is employed to compute the inertial terms without any iteration. This new transformation is implemented in NASTRAN using a DMAP sequence alter. Numerical examples using a cantilever beam illustrate the necessary condition for allowing a large number of additional terms in the proposed series correction of Guyan reduction. A practical example of a large model of the Plasma Motor Generator module to be flown on a Delta launch vehicle is also presented.*

1. Introduction: The dynamic analysis of complicated structures often produces large finite element models. In some instances, the automated computer procedures to generate finite element meshes also lead to large models. These highly refined models are really a byproduct of the use of model generating software and they may not be needed for accuracy. A common approach to reduce the size of the eigenvalue problem for structural dynamics applications is Guyan reduction. This approximate method finds its place among other applications also. For the purposes of cost-effectiveness, Guyan reduction is employed in Coupled Loads Analysis using substructuring techniques. In the experimental modes analysis, analytical selection of retained degrees of freedom for Guyan reduction is used as a guide to select accelerometer locations on the test article. Mass weighted orthogonality computations between the test and analytical modeshapes are performed using Guyan reduction.

While Guyan reduction [1] is exact in static applications, it introduces approximations in structural dynamics. The correct relationship between the retained and omitted degrees of freedom can be expressed in the form of a series. The Guyan reduced mass and stiffness matrices, available in explicit form, are used to compute the series terms approximately. The Guyan reduced matrices provide the best possible solution without requiring any further iterations. The condition for convergence of the series and the relationship of this series transformation to the improved reduced system (IRS) introduced by O'Callahan [2]

are examined in this paper.

2. Theory: The eigenvalue problem from the structural dynamic analysis is given as

$$K\mathbf{u} = \lambda M\mathbf{u} \tag{1}$$

Eq. (1) can be written in the partitioned form as,

$$\begin{bmatrix} K_{aa} & K_{ao} \\ K_{oa} & K_{oo} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_a \\ \\ \mathbf{u}_o \end{Bmatrix} = \lambda \begin{bmatrix} M_{aa} & M_{ao} \\ M_{oa} & M_{oo} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_a \\ \\ \mathbf{u}_o \end{Bmatrix} \tag{2}$$

where $\mathbf{u}_a$ represents the eigenvector of the retained degrees of freedom and $\mathbf{u}_o$ the eigenvector of the degrees of freedom omitted in the Guyan reduction. M_{ij} and K_{ij} are the corresponding submatrices of the mass and stiffness matrices respectively and λ is the eigenvalue. The second partition of Eq. (2) can be written separately as

$$(K_{oa} - \lambda M_{oa})\,\mathbf{u}_a + (K_{oo} - \lambda M_{oo})\,\mathbf{u}_o = 0 \tag{3}$$

Expanding the vector $\mathbf{u}_o$ in terms of $\mathbf{u}_a$ from Eq. (3),

$$\begin{aligned} \mathbf{u}_o &= -\left(K_{oo} - \lambda M_{oo}\right)^{-1} \left(K_{oa} - \lambda M_{oa}\right) \mathbf{u}_a \\ &= -\left(I - \lambda K_{oo}^{-1} M_{oo}\right)^{-1} \left(K_{oo}^{-1} K_{oa} - \lambda K_{oo}^{-1} M_{oa}\right) \mathbf{u}_a \end{aligned} \tag{4}$$

Guyan reduction transformation leaves out the frequency dependent terms in Eq. (3). Hence, the regular Guyan reduction transformation becomes,

$$\mathbf{u}_o = -K_{oo}^{-1} K_{oa} \mathbf{u}_a \tag{5}$$

If the condition for convergence (Section 4.) is satisfied, the inverse of $\left(I - \lambda K_{oo}^{-1} M_{oo}\right)$ can be expanded in Neumann series as,

$$\left(I - \lambda K_{oo}^{-1} M_{oo}\right)^{-1} = I + \lambda K_{oo}^{-1} M_{oo} + \lambda^2 \left[K_{oo}^{-1} M_{oo}\right]^2 + \ldots \tag{6}$$

Using Eq. (6) in Eq. (4) and simplifying the terms yields,

$$\mathbf{u}_o = -\left[K_{oo}^{-1} K_{oa} + B\lambda + AB\lambda^2 + A^2 B\lambda^3 + \ldots\right] \mathbf{u}_a \tag{7}$$

where

$$A = K_{oo}^{-1} M_{oo} \quad \text{and} \quad B = K_{oo}^{-1} M_{oa} - A K_{oo}^{-1} K_{oa} \tag{8}$$

The exact relationship between $\mathbf{u}_o$ and $\mathbf{u}_a$ in Eq. (7) involves nonlinear terms of the unknown eigenvalues (λ). A practical approximation to compute these terms in Eq. (7) can be made from regular Guyan reduction by taking,

$$K_r \mathbf{u}_a \cong \lambda M_r \mathbf{u}_a \tag{9}$$

where K_r and M_r are Guyan reduced stiffness and mass matrices respectively and are given explicitly as,

$$\begin{aligned} K_r &= K_{aa} - K_{ao}K_{oo}^{-1}K_{oa} \\ M_r &= M_{aa} - M_{ao}K_{oo}^{-1}K_{oa} - K_{oa}K_{oo}^{-1}M_{oa} + K_{ao}K_{oo}^{-1}M_{oo}K_{oo}^{-1}K_{oa} \end{aligned} \tag{10}$$

From Eq. (9), it is seen that,

$$\lambda \mathbf{u}_a = M_r^{-1}K_r\mathbf{u}_a \tag{11}$$

Using Eq. (11) repeatedly, it can be shown that,

$$\begin{aligned} \lambda^2\mathbf{u}_a &= \left(M_r^{-1}K_r\right)^2\mathbf{u}_a \\ &\vdots \\ \lambda^i\mathbf{u}_a &= C^i\mathbf{u}_a, \quad C = M_r^{-1}K_r \end{aligned} \tag{12}$$

Substituting Eq. (12) into Eq. (7), the relationship between $\mathbf{u}$ in Eq. (1) and $\mathbf{u}_a$ becomes,

$$\mathbf{u} = T\mathbf{u}_a \tag{13}$$

where

$$T = \begin{bmatrix} I \\ -K_{oo}^{-1}K_{oa} + \sum_{i=1,2,..} A^{i-1}BC^i \end{bmatrix} \tag{14}$$

By applying the relation between $\mathbf{u}$ and $\mathbf{u}_a$ in Eq. (13), the new improved matrices from series reduction can be obtained as,

$$\overline{K} = T^TKT \quad \text{and} \quad \overline{M} = T^TMT \tag{15}$$

It is interesting to note that M_{oa} vanishes for lumped formulations of the mass matrix. Taking the value of i to be unity, the transformation in Eq. (14) reduces to

$$T = \begin{bmatrix} I \\ -K_{oo}^{-1}K_{oa} + BC \end{bmatrix} \tag{16}$$

which is the improved reduced system (IRS) proposed by O'Callahan [2].

3. DMAP Alter: A rigid format alter for dynamic analysis in NASTRAN has been developed to incorporate the improved Guyan reduction with the series terms. A parameter called GOPT is used to choose the number of correction terms. The alter listing is also provided in this section.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$        CSA/NASTRAN ALTER FOR IMPROVED GUYAN REDUCTION
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
RFINSERT SMP2 $
PARAM //C,N,NOP/V,Y,GOPT=-1 $
PARAM //C,N,SUB/V,N,GOUT/V,Y,GOPT/C,N,2 $
COND LGOPT,GOPT $
UPARTN USET,MFF/MAAB,MOA,,MOO/*F*/*A*/*O* $
FBS LOO,,MOO/AMAT/1 $
FBS LOO,,MOA/BMAT1/1 $
MPYAD AMAT,GO,BMAT1/BMAT $
SOLVE MAA,KAA/CMAT/1 $
$
MPYAD BMAT,CMAT,/SUM $
COND OUT,GOUT $
MATMOD SUM,,,,,/PRDT,/13 $
LABEL LOOPTOP $
EQUIV SUM,SUM1/NEVER $
EQUIV PRDT,PRDTX/NEVER $
SMYPAD AMAT,PRDT,CMAT,,,/PRDTX/3 $
ADD SUM,PRDTX/SUM1 $
EQUIV SUM1,SUM/ALWAYS $
EQUIV PRDTX,PRDT/ALWAYS $
REPT LOOPTOP,GOUT $
LABEL OUT $
ADD GO,SUM/GONE $
SMP2 USET,GONE,MFF/MAA $
SMP2 USET,GONE,KFF/KAA $
LABEL LGOPT $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

4. Validity of Guyan Reduction: The inverse of the matrix $\left[I - \lambda K_{oo}^{-1} M_{oo}\right]$ in Eq. (4) can be expanded as a converging Neumann series only if all the eigenvalues of $\lambda K_{oo}^{-1} M_{oo}$ are less than unity. In other words, the Guyan reduction is valid only for those frequencies less than the smallest frequency of the eigenvalue problem formed out of the omitted degrees of freedom. The effect of violating this condition will be scrutinized in the next section.

5. Demonstration Examples:

5.1 Uniform Cantilever: The first example is concerned with a cantilevered bar clamped at one end. The relevent structural parameters are taken to be the modulus of elasticity (E) being equal to 30×10^6 *psi*, weight density (ρg) as 0.2839 lb/in^3, area of cross section as 1 in^2 and the length of the bar (L) as 72 in.

The characteristic equation of this cantilever is $\cos \sqrt{\rho/E}\omega L = 0$ from which the theoretical natural frequencies can be computed. The cantilever is divided into twenty finite

elements. The retained degrees of freedom for Guyan reduction are the axial displacements at the free end and at two successive nodes. The reduction transformation includes n as the number of additional series terms. The natural frequencies from improved Guyan reduction for different values of n are listed in Tables 1 through 5.

Table 1. Cantilever Frequency Comparisons ($n = 0$)

Standard Guyan Reduction

Mode Number	Theoretical Frequency (Hz)	Computed Frequency (Hz)	Error %
1	7.012E2	7.428E2	5.926E0
2	2.104E3	7.562E3	2.595E2
3	3.506E3	1.655E4	3.722E2

Table 2. Cantilever Frequency Comparisons ($n = 1$)

Mode Number	Theoretical Frequency (Hz)	Computed Frequency (Hz)	Error %
1	7.012E2	7.012E2	8.014E-2
2	2.104E3	2.583E3	2.279E1
3	3.506E3	1.390E4	2.964E2

Table 3. Cantilever Frequency Comparisons ($n = 2$)

Mode Number	Theoretical Frequency (Hz)	Computed Frequency (Hz)	Error %
1	7.012E2	7.011E2	-2.169E-2
2	2.104E3	2.239E3	6.440E0
3	3.506E3	7.043E3	1.009E2

Table 4. Cantilever Frequency Comparisons ($n = 3$)

Mode Number	Theoretical Frequency (Hz)	Computed Frequency (Hz)	Error %
1	7.012E2	7.012E2	-2.170E-2
2	2.104E3	2.167E3	3.003E0
3	3.506E3	3.879E3	1.063E1

Table 5. Cantilever Frequency Comparisons ($n = 4$)

Mode Number	Theoretical Frequency (Hz)	Computed Frequency (Hz)	Error %
1	7.012E2	7.010E2	-3.235E-2
2	2.104E3	2.120E3	7.937E-1
3	3.506E3	3.680E3	4.950E0

The accuracy of the computed frequencies is improved by taking into account the higher order correction terms. However, when $n \geq 6$, the reduced mass matrix is no longer positive definite and the eigenvalue solution process breaks down. This limitation of adding a finite number of correction terms can be explained by the fact that the third frequency of the overall structure exceeds the lowest frequency of the omit set (O-set) system thus violating the convergence criterion for Guyan reduction.

Another cantilever example is constructed by assuming that the three elements near the free end are made up of a material with $E = 30 \times 10^4$ *psi* instead of steel. By retaining the same degrees of freedom as in the previous example of all steel construction, it becomes possible to add an almost limitless number of correction terms. This is because there is no overlap between the frequency spectrum of the first three modes of the full system and that of the O-set system.

5.2 Plama Motor Generator (PMG): This example comes from the modal testing and finite element analysis of the PMG Far End Package (Figure 1). The PMG experiment is a payload on a Delta II 7925 launch vehicle. The mission is scheduled to take place in July 1993.

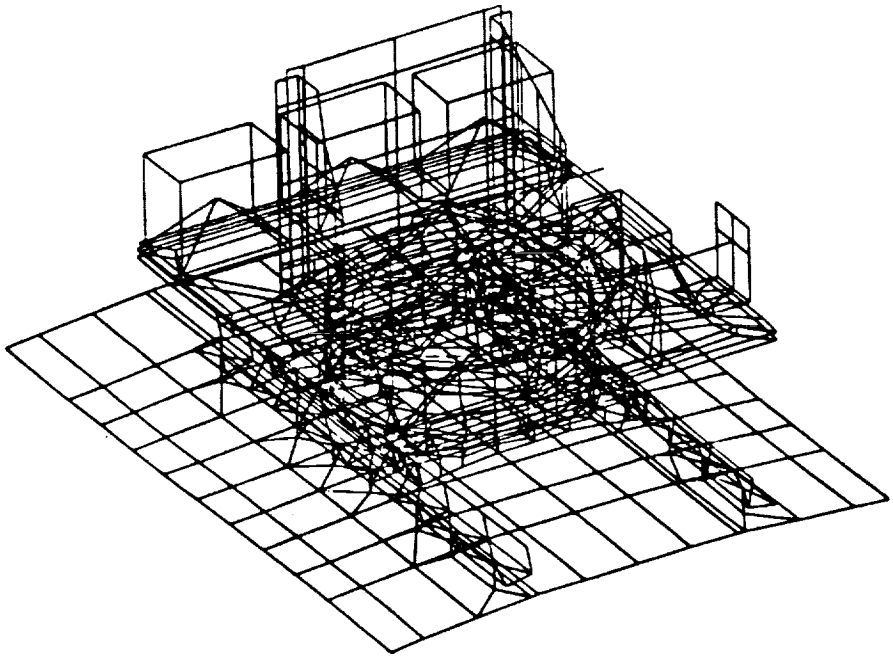


Figure 1. PMG Far End Package

The analysis set degrees of freedom correspond to the accelerometer locations used in the modal survey test. The improved Guyan reduction is performed with different n on the PMG Far End Package model. The computed frequencies are compared with those of the full model which are taken as the reference solution and the results are listed in Tables 6 through 8. Several frequencies that were not found by the standard Guyan reduction start to reappear by adding the correction terms.

Table 6. PMG Frequency Comparisons ($n \stackrel{!}{=} 0$)

Mode Number	Reference Frequency (Hz)	Computed Frequency (Hz)	Error %
1	56.32	56.39	0.13
2	84.46	84.66	0.23
3	100.66	101.20	0.53
4	118.19	119.58	1.17
5	159.46	160.48	0.63
6	170.06	—	—
7	185.19	—	—
8	215.16	220.65	2.55
9	217.65	224.09	2.95
10	228.36	236.73	3.56
11	234.52	—	—
12	243.43	256.55	5.38
13	264.53	—	—
14	299.03	—	—
15	305.16	330.53	8.31

Table 7. PMG Frequency Comparisons ($n = 1$)

Mode Number	Reference Frequency (Hz)	Computed Frequency (Hz)	Error %
1	56.32	56.32	0.00
2	84.46	84.46	0.45E-4
3	100.66	100.66	0.59E-3
4	118.19	118.20	0.46E-2
5	159.46	159.46	0.16E-2
6	170.06	171.65	0.92

Table 8. PMG Frequency Comparisons ($n = 2$)

Mode Number	Reference Frequency (Hz)	Computed Frequency (Hz)	Error %
1	56.32	56.82	0.89
2	84.46	85.52	1.24
3	100.66	101.12	0.45
4	118.19	118.31	0.09
5	159.46	160.32	0.53
6	170.06	170.18	0.064
7	185.19	185.29	0.054
8	215.16	215.19	0.016
9	217.65	217.71	0.026
10	228.36	228.61	0.10
11	234.52	234.60	0.035
12	243.43	242.79	-0.26
13	264.53	264.65	0.043
14	299.03	299.59	0.17
15	305.16	305.60	0.14

6. Conclusion:

A noniterative procedure to enhance the standard Guyan reduction with a series of terms has been presented. In practice, it may be possible to add only a finite number of the correction terms as demonstrated by the NASTRAN examples.

7. References:

1. Guyan, R. J., Reduction of Stiffness and Mass Matrices, AIAA Journal, Vol. 3, 1965.

2. O'Callahan, J. C., A Procedure for an Improved Reduced System Model, Procedings of the Seventh International Modal Analysis Conference, 1989.

DESIGN OPTIMIZATION STUDIES USING COSMIC NASTRAN

57-39
190577
N94-17834

S. M. Pitrof, G. Bharatram, V. B. Venkayya
Wright Laboratory
Wright-Patterson AFB OH 45433-7552

Summary

The purpose of this study is to create, test and document a procedure to integrate mathematical optimization algorithms with COSMIC NASTRAN. This procedure is very important to structural design engineers who wish to capitalize on optimization methods to ensure that their design is optimized for its intended application. The OPTNAST computer program was created to link NASTRAN and design optimization codes into one package. This implementation was tested using two truss structure models and optimizing their designs for minimum weight, subject to multiple loading conditions and displacement and stress constraints. However, the process is generalized so that an engineer could design other types of elements by adding to or modifying some parts of the code.

Introduction

Since the advent of NASTRAN during the early 70's, engineers have found many applications of finite element analysis in diverse fields. Its popularity, which is still growing, has spawned many commercial and research programs and they are available on just about every kind of computer available on the market. The parallel development of graphics interfaces, which started as pre- and post-processors to finite element programs, have further stimulated fascinating applications in the analysis of mechanical components, built-up structures, fluid-structure interaction problems, thermal and heat transfer analysis, acoustics and other engineering analyses. The reliability of finite element analysis is increasingly attributed to the graphical aids. They are the means for model error correction, display of analysis results such as displacements, mode shapes (including animation), color coded displays of stresses and strains, etc. With shrinking budgets and increasing competition for market share, the industry is groping for ways to cut product development costs and reduce development time from concept to market. Analysis tools such as NASTRAN offer challenging opportunities for rapid parametric studies at minimal cost. Adept use of these tools is the key to improving quality and reducing cost of new products. These two aspects are the most important ingredients for market leadership.

Having realized the many advantages of finite element analysis during the 70's, engineers have embarked upon the development of even more ambitious integrated design systems in the name of computer aided engineering (CAE). The basic elements of these multidisciplinary systems are finite element analysis and mathematical optimization (nonlinear programming) algorithms coupled by sensitivity analysis. The sensitivity analysis is an extension of finite element analysis through first order approximations. These integrated systems take full advantage of the ever improving capabilities of modern digital computers and provide significant reductions in product development costs and time. The objective of this paper is to show how COSMIC NASTRAN, which is basically an analysis tool, can be coupled to a nonlinear programming package to obtain an optimized structure. Although the single discipline analysis architecture of NASTRAN presents numerous difficulties, it is possible to achieve objectives of optimization to a limited extent. The bridge between the analysis and optimization is the sensitivity analysis and the procedure outlined in Reference 1 is used in this implementation.

The next section provides a brief introduction to optimization theory and sensitivity analysis, followed by some details of the implementation using COSMIC NASTRAN. This is followed by discussion of the results gained from this implementation as applied to simple truss problems.

Theory

The optimization problem is generally posed as follows:

Minimize an objective function:

$$F(\underset{\sim}{x}) = F(x_1, x_2, \ldots x_a)$$

Subject to a set of constraints:

$$z_i(\underset{\sim}{x}) = z_i(x_1, x_2, \ldots x_a) \leq \bar{z}_i$$

$$z_j(\underset{\sim}{x}) = z_j(x_1, x_2, \ldots x_a) = \bar{z}_j$$

$$\underset{\sim}{x}^l \leq \underset{\sim}{x} \leq \underset{\sim}{x}^u$$

F is the user defined objective function, while $\underset{\sim}{x}$ is the vector of design variables. The first set of constraints, z_i, is the inequality constraints. The second set z_j is the equality constraints. The third set is the constraints on the variables (upper and lower bounds) themselves. The weight of the structure is the objective function addressed in this paper while the constraints are on the displacements and stresses. The variables in the structural optimization problem described in this paper are the cross-

sectional areas of the rods, but could instead be thicknesses of the plates or some other design parameter.

The constraints are non-linear functions of the variables and thus the problem comes under the category of nonlinear programming. The iterative solution of the linear or nonlinear programming problems can be written as:

$$\underset{\sim}{x}^{v+1} = \underset{\sim}{x}^{v} + \tau \underset{\sim}{D}$$

where $\underset{\sim}{x}^{v}$ and $\underset{\sim}{x}^{v+1}$ are the variable vectors in two consecutive cycles, $\underset{\sim}{D}$ (∇F, ∇Z) is the travel direction or perturbation, and τ is the step size. The travel direction in most gradient-based solutions is based on the objective and constraint function gradients (∇F and ∇Z).

So basically, the steps involved in the solution of the nonlinear programming problem are as follows:

1. Initial solution $\underset{\sim}{x}$
2. Function evaluation
3. Selection of active constraints
4. Gradient evaluation
5. Determine the travel direction $\underset{\sim}{D}$
6. Determine the step size τ .
7. Check for the optimality conditions.
8. Repeat the steps until the conditions are satisfied.

Gradient computations are as outlined in Reference 1. CONMIN, a nonlinear programming package based on the modified method of feasible directions, is used as the optimizer (Reference 2).

Implementation

As previously discussed, there is potential for considerable benefit in performing structural design optimization studies using NASTRAN. However, integrating optimization algorithms with NASTRAN has been a daunting proposition. The effort required to develop a fully integrated structural design optimization package is so extensive that only through intensive, dedicated efforts such as the Air Force's Automated STRuctural

Optimization Program (ASTROS) program can finite element analysis codes and mathematical optimization algorithms be interfaced into a system capable of performing structural design. A true integrated package such as ASTROS consists of one executable program, with all capabilities built into it. Another approach, which we will discuss in this paper, is to synthesize separate executable files with a shell script program run by the computer's operating system. The script program calls multiple executable files and performs some rudimentary computations and data processing activities. In the past few years two phenomena have emerged to make our task of implementing optimization in NASTRAN far more realizable.

The first is the emergence of code written in subroutine form to compute values needed as inputs to optimization algorithms such as constraint values and constraint sensitivities. Optimization algorithms need to specify a design problem as an objective function to be maximized or minimized. As design variable values change, the objective function value changes. The algorithm also requires that bounds on the problem are placed. These bounds take form as constraint values and design variable upper and lower bounds. Much of the information required by optimization algorithms is very simple and straightforward to compute. Some values such as initial design variable values, design variable value upper and lower bounds, and constraint limits are left to the user to define. Other values such as objective function values and constraint values are fairly simple to compute but require information about the structure such as geometry and response to loading. Of significantly greater difficulty to compute are objective function and constraint sensitivities. Sensitivity values, which are defined as the first derivative of the objective and constraint functions, tell the optimizer which direction in design space to move. Recently, programs in subroutine form to compute such values have become more available (exemplified in Reference 1) to calculate constraint sensitivities for NASTRAN elements.

The second phenomenon is the emergence of open computer operating architectures. Cosmic NASTRAN has in the past been available on proprietary computer architectures such as CDC/CYBER and VAX/VMS. As Unix systems are becoming more available, NASTRAN is migrating to these new machines in order to take advantage of open systems. This environment is especially amenable to programmers who wish to integrate stand-alone programs into a package but either cannot or choose not to rewrite stand-alone programs in subroutine format and link operation by a main driver program. Since we have programs such as NASTRAN to perform structural analyses , programs such as CONMIN to perform optimization studies, and many miscellaneous programs to formulate input values required for optimization from output values from NASTRAN, Unix provides us with the necessary capability to synthesize these programs into one system capable of performing structural optimization tasks.

The OPTNAST computer program was created to demonstrate the feasibility of integrating NASTRAN with optimization methods in the context of structural design. OPTNAST, which capitalizes on previously written optimization code and the Unix operating system, consists of several fortran programs and a Unix shell script program. The Unix c-shell script was written to perform a loop operation between the analysis program (NASTRAN) and the optimizer (CONMIN). In order to use the script the user must obey some basic rules regarding his design problem. These rules are imposed on the user in order to simplify the code development process. The restrictions are as follows:

- No free format
- Only one material card
- All elements will be designed
- Constraints will be applied to all elements/nodes
- All load cases will be designed; limit of 5 load cases

With more extensive code development, any of these restrictions can be removed. However, our intent is to develop a reasonably practical methodology to conduct optimization with NASTRAN and thus some restrictions are acceptable.

There are two input files required by the OPTNAST program. They are a standard NASTRAN input file (e. g. tenbar.nid) and a file of optimization parameters (e. g. tenbar.opt). The input file must obey the previously discussed restrictions and must also include the following statements:

- Request for OUTPUT2 file with KELM matrix (for use in gradient computations)
- Request for punch file with displacement and/or stress data (for use in constraint calculations)

The optimization parameter file must contain the following:

- New CONMIN parameters to override defaults (if any are desired)
- Number of and values for displacement and stress constraints

Examples of each are contained in Appendices 1 and 2.

Once the user has properly prepared the NASTRAN input file and the optimization parameter file, the user is ready to run the OPTNAST program (Figure 1). The OPTNAST program consists of a Unix script (Appendix 3) file that calls the executable programs and processes the data shared by the executables. There are three executable files called by OPTNAST. The first is PREPARE, which preprocesses the bulk data file. The second is NASTRAN, and the third is COSOPT, which performs all of the optimization computations (Appendix 4). The OPTNAST script performs the following operations:

- Reads the name of the input file
- Processes the input file to include load cases to calculate virtual load vector response (for gradient calculations)
- Submits the problem to NASTRAN to calculate initial structural design response to the applied loads
- Sends the data to the COSOPT program to:

(1) calculate constraint and objective function and gradient values
(2) submit to CONMIN for optimization
(3) return a new NASTRAN input file if design has not converged or a converge flag if it has

- Loop back and submit new input file to NASTRAN to continue optimization task
- Continue looping until optimum is reached or maximum number of 16 iterations is reached

While the OPTNAST program is not an integrated package, but rather a collection of executables driven by a script file, it is fully capable of performing all tasks necessary to solve the optimization problem.

Results and Discussion

The OPTNAST program was used to perform design optimization studies on two structural models, each with varying constraint values and load cases. The first model, the Ten Bar Truss (Figure 1) was modeled with the properties as illustrated in the NASTRAN input file example (Appendix 1). This problem was solved with six different conditions, with minimization of structure weight being the objective in each case. The first case featured 2.0" displacement constraints applied to all grid points. The second case featured 25000 psi stress constraints (both tensile and compressive) on each element. The third case synthesized both the first two cases. The fourth case featured stress constraints with two separate load cases applied. The fifth case was identical to the second case except that no linear approximations were made during the redesign phase (NASTRAN was called to recalculate structural response after each iteration). The sixth case was again identical to the second, except that the initial design variable values are set to minimum gauge. This is what is described as an infeasible design because all constraints are violated.

The second structural model designed was a Two Hundred Bar Truss (Figure 3). The objective of this model is to provide an example of a large structure in order to indicate feasibility of designing a large model. This structure was solved with stress constraints applied to each element and with two separate load cases. Since there are two hundred elements and two load cases, this design model includes two hundred design variables and four hundred constraints.

Each of the previously described models was run with the OPTNAST program, and results are provided to compare with those provided by the ASTROS program. Since ASTROS input is generally compatible with NASTRAN and since ASTROS uses a similar optimization algorithm, approximation concepts and gradient calculations, results gained from each code should be comparable. This comparison is bourne out when viewing the final results tabulated in Table 1. The results show that for any design the final design's optimal weight for each method agree to within one percent. One obvious penalty is that the amount of time required is much less with an integrated package like ASTROS. Improvements to the

OPTNAST program can be made to improve efficiency, but an integrated package with a centralized database like ASTROS benefits from inherently more efficient methods of processing, storing and sharing data between modules. It should also be noted that the timing summary for the OPTNAST program is only an approximation since the code was not included to keep track of the actual time spent.

Concluding Remarks

This study has proved the feasibility of conducting optimization studies with NASTRAN. The OPTNAST program generated for this study can be used for designing truss structures with displacement and stress constraints. As many as five different load cases can be considered with the program. The program can achieve optimum designs very similar to integrated design optimization packages such as ASTROS, but a computational performance penalty is inherent and unavoidable. Still, this method is very attractive when integrated packages do not offer the necessary capabilities, such as element types or constraints that the user needs to design for. As a result, this is a viable alternative when the user has highly specialized design needs.

References

1. Tischler, V. A.; Venkayya, V. B.: Sensitivity Analysis and Optimization Issues in NASTRAN, 1991 NASTRAN Colloquium

2. Vanderplaats, Garret N.: CONMIN - A Fortran Program for Constrained Function Minimization User's Manual, NASA TMX-62282, Ames Research Center, August 1973

3. COSMIC NASTRAN User's Manual, NASA SP-222(08), June 1986

Tables

Table 1: Results

Case		OPTNAST			ASTROS		
Model Name	Constra-ints	Weight (lbs)	Iteration cycles	Clock (min)	Weight (lbs)	Iteration cycles	Clock (min)
10 Bar Truss	Disp	5024	12	12:00	5102	12	1:15
10 Bar Truss	Disp & Stress	5066	10	10:00	5104	12	1:41
10 Bar Truss	Stress Const	1594	14	14:00	1594	18	2:31
10 Bar Truss	2xLoads, Stress	1741	9	9:00	1738	15	1:26
10 Bar Truss	Stress, No Approx	1609	144	Hrs	-	-	-
10 Bar Truss	Stress, Infeas.	1594	13	13:00	1593	16	1:30
200 Bar Truss	2xLoads, Stress	98.62	12	5 Hrs	98.75	6	8:47

Figures

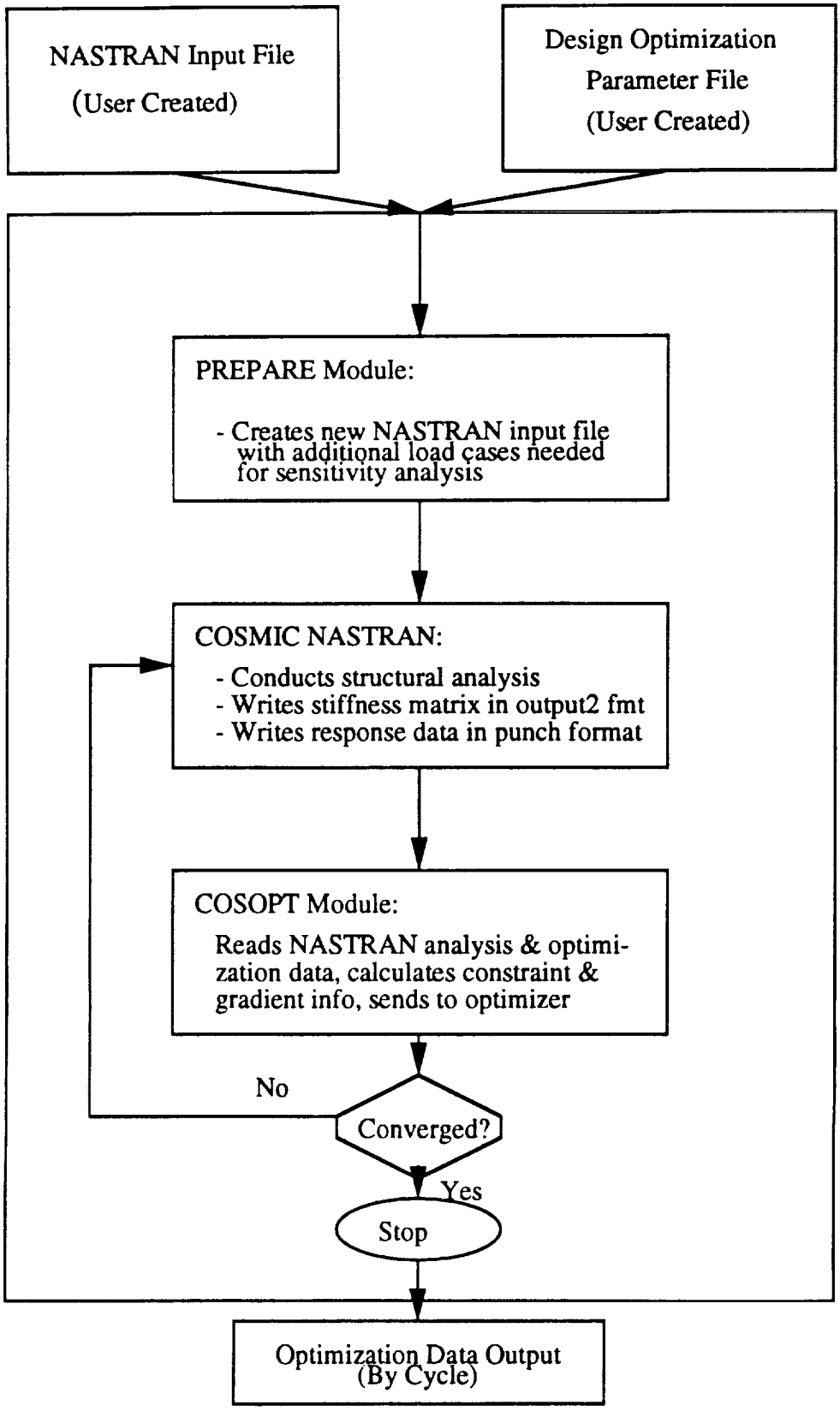


Figure 1: OPTNAST Program

apshot

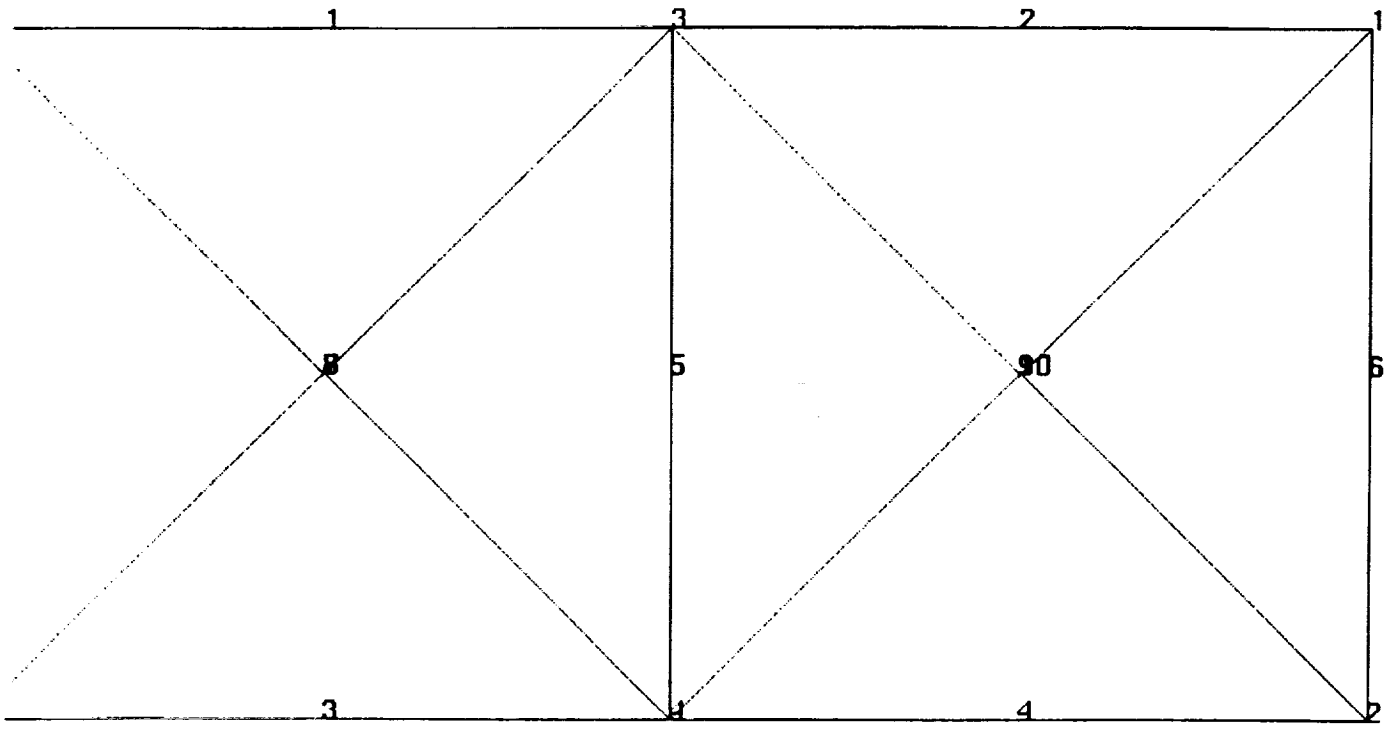


Figure 2: Ten Bar Truss

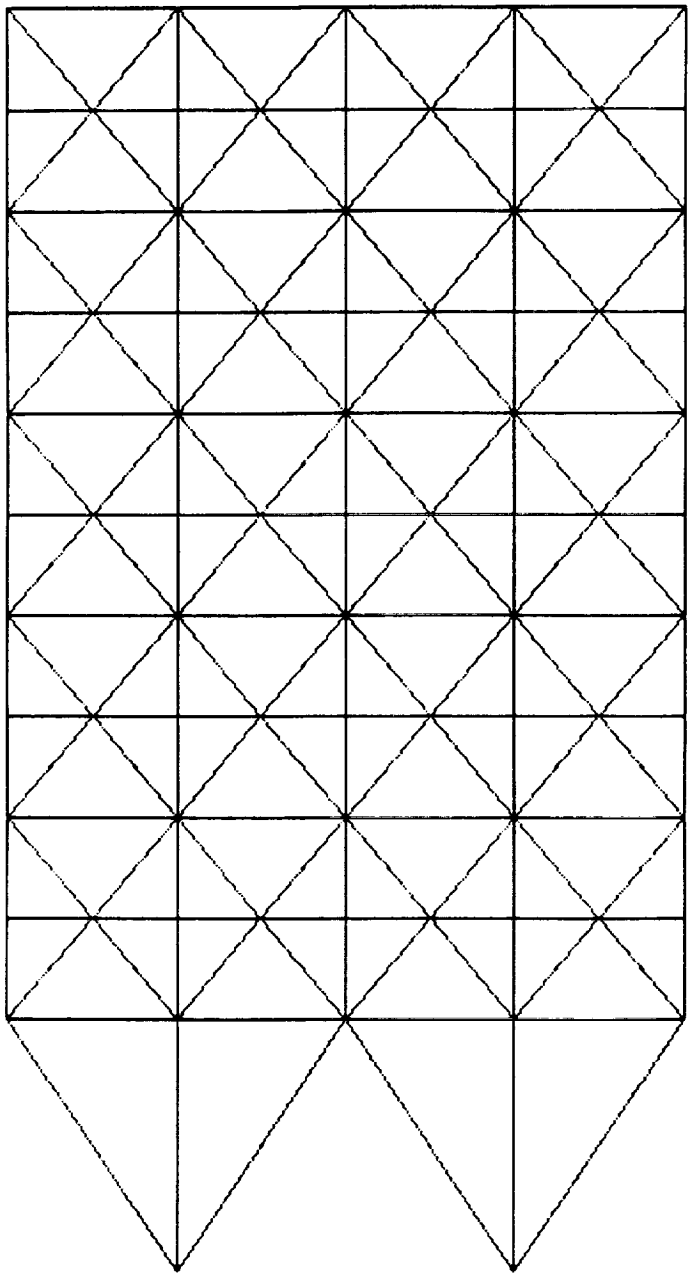


Figure 3: Two Hundred Bar Truss

```
ID TENB,TENB
SOL 1,0
TIME 50
ALTER 37 $
OUTPUT2 KELM//-1/15/ V,N,Z $
OUTPUT2,,,,//-9/15 $
ENDALTER $
CEND
TITLE = TEN BAR TRUSS
DISP(PRINT,PUNCH)=ALL
STRESS(PRINT,PUNCH)=ALL
SPC = 1
SUBCASE 1
LOAD = 1
BEGIN BULK
$
$              TEN BAR TRUSS MODEL
$              FROM SCHMIT, L.A., JR. AND MIURA, H., " APPROXIMATION
$                      CONCEPTS FOR EFFICIENT STRUCTURAL SYNTHESIS ",
$                      NASA CR-2552, MARCH 1976.
$
GRID      1                720.0    360.0    0.0
GRID      2                720.0     0.0     0.0
GRID      3                360.0    360.0    0.0
GRID      4                360.0     0.0     0.0
GRID      5                  0.0    360.0    0.0
GRID      6                  0.0     0.0     0.0
CROD      1        1       3        5
CROD      2        2       1        3
CROD      3        3       4        6
CROD      4        4       2        4
CROD      5        5       3        4
CROD      6        6       1        2
CROD      7        7       4        5
CROD      8        8       3        6
CROD      9        9       2        3
CROD     10        10      1        4
PROD      1         2      30.0
PROD      2         2      30.0
PROD      3         2      30.0
PROD      4         2      30.0
PROD      5         2      30.0
PROD      6         2      30.0
PROD      7         2      30.0
PROD      8         2      30.0
PROD      9         2      30.0
PROD     10         2      30.0
$
MAT1      2      1.E+7             0.3      0.1       25000.0
$
SPC1,     1,   123456,  5,   6
SPC1,     1,   3456,    1,   THRU,  4
$
FORCE,    1,   2,  ,  -1.E5,   0.0,     1.0,     0.0
FORCE,    1,   4,  ,  -1.E5,   0.0,     1.0,     0.0
$
ENDDATA
```

Appendix 1: NASTRAN Input File

```
$ AN EXAMPLE PROBLEM
INDMIN=0
0.10
$ PRINT CONTROL
 IPRCTL=3
$ DISPLACEMENT CONSTRAINT
LMTDSP=2
6,-2.0   2   1
  -2.0   2   2
  -2.0   2   3
  -2.0   2   4
  -2.0   2   5
  -2.0   2   6
 NZLMIT=4
 IPRINT=1
 FXMIN=1.0E+10
 ITERT=40
 XMIN=.1
 XMAX=1000.0
```

Appendix 2: Optimization Parameter Input File

```
# Unix c-shell script to optimize rod structure for displacement
# and stress constraints using NASTRAN to derive structural response
# quantities (displacements, stresses, K matrix), CONMIN optimization
# algorithm for optimization and assorted routines to calculate
# objective function, constraint values and sensitivities (sensitivity
# analysis uses virtual load vector method
#
# Inputs to program are NASTRAN input deck (no free format) <filename.nid> and
# optimization parameter file <filename.opt>
#
# get model name if not provided
if ($1 == "") then
echo 'model name?'
set a = $<
else
set a = $1
endif
# check to see if optimization parameter file exists
if (! -e $a.opt) then
echo "RUN REQUIRES OPTIMIZATION PARAMETERS ($a.opt)
exit
endif
echo "1.0" >fort.85              #initialize last obj fn value to 1.0
cp $a.nid $a.nid.old             #save old input
cp $a.opt fort.4                 #get optimization date
cp $a.nid fort.55                #copy input to unit 55
prepare <$a.nid >$a.out          #add virtual load vectors to NASTRAN input
rm $a.out $a.nid
mv fort.65 $a.nid
# build script to execute cosmic
echo "c" >cosfeed
echo $a >>cosfeed
echo "o" >>cosfeed
echo "i" >>cosfeed
echo "y" >>cosfeed
set it = 0
#begin loop
while ($it < 16)                 #maximum 16 iterations
cp $a.nid fort.55
@ it = $it + 1                   #counter
cosmic <cosfeed >$a.out          #execute cosmic interactively
#prepare for optimization segment
#cp $a.nid fort.55               #copy input file to unit 55
cp $a/PCH fort.25                #punch file to unit 25
cp $a/INP1 fort.15               #output2 file to unit 15
rm -rf $a
cosopt <fort.55 >$a.opt.it$it    #submit to optimization program
if ( -e fort.65 ) mv fort.65 $a.nid
set loop = `cat fort.75`
if ( $loop == "0" ) set it="16" #if optimization converged end loop
end
rm cosfeed fort.15 fort.25 fort.55 fort.75 fort.4 fort.85
```

Appendix 3: OPTNAST Unix Shell Script

```
      PROGRAM COSOPT
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-----------------------------------------------------------------------
C     Program to submit NASTRAN output to CONMIN optimization algorithn
C     for rod structures with displacement and stress constraints
C-----------------------------------------------------------------------
      EXTERNAL SETFUN
C-----------------------------------------------------------------------
      INCLUDE 'cosopt.inc'
C-----------------------------------------------------------------------
      COMMON/CNMN1/DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,
     +ALPHAX,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,
     +NSCAL,LINOBJ,ITMAX,ITRM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER
C-----------------------------------------------------------------------
      SAVE/FUNPAR/
      COMMON/FUNPAR/FXMIN,XL,XU,NZLMIT,ITERT,IPRINT1
C-----------------------------------------------------------------------
C  Thickness (of membrane elements or area of bars--input)
C
      DIMENSION TH( MAXELM )
      SAVE   /ANLYZ1/
      COMMON /ANLYZ1/ TH, MEMBS, JOINTS, MM, NFI
C-----------------------------------------------------------------------
C
C  Index to elements' material properties
C
      INTEGER MYOUNG( MAXELM )
C
C  Material properties
C
      DIMENSION YOUNGM( MAXMTL ), POISON( MAXMTL ), RHO1( MAXMTL )
C
C  Allowable Stresses
C
      DIMENSION ALSTRS( 3, MAXMTL )
C
      SAVE   /ANLYZ2/
      COMMON /ANLYZ2/ EEE, PMU, RHO, YOUNGM, POISON, RHO1, MYOUNG,
     +                ALSTRS, NMAT, MSSTRS
C-----------------------------------------------------------------------
      INTEGER NNODES( MAXELM )
C
C  Node number connectivities for each element
C
      INTEGER MA( MAXELM ), MB( MAXELM ), MC( MAXELM ), MD( MAXELM )
C
C  Nodal coordinates for each joint
C
      DIMENSION X( MAXJNT ), Y( MAXJNT ), Z( MAXJNT )
C
      SAVE   /ANLYZ3/
      COMMON /ANLYZ3/ NNODES, MA, MB, MC, MD, X, Y, Z, INCHES
C-----------------------------------------------------------------------
C
C  Degree of freedom numbers for restrained nodes (boundary conditions)
C
      DIMENSION IBND( MAXBND )
C
C  Number of load components for each loading condition
C
```

```
      DIMENSION NJLODS( MAXLOD )
C
C  Displacement and force resultants for each degree of freedom and
C   and loading condition
C
      DIMENSION DR( NNMAX,MAXLOD ), FR( NNMAX,MAXLOD )
C
C  Stiffness and mass matrices
C
      DIMENSION SK( MAXSK ), GM( MAXSK )
C
C  Pointers to diagonal elements in stiffness matrix, SK;
C  Row number for first nonzero element in each Column of SK
C
      DIMENSION IDIAG( NNMAX ), ICOL( NNMAX )
C
      SAVE   /ANLYZ4/
      COMMON /ANLYZ4/ IBND, NJLODS, FR, DR, SK, IDIAG, ICOL, GM,
     +                NBNDRY, NN, KIPS, NR, NONZRO
C-----------------------------------------------------------------------
      SAVE   /ANLYZ5/
      COMMON /ANLYZ5/ LOADS
C-----------------------------------------------------------------------
C
C  Element Area (size) Minimum and Maximums,
C  Variable Bounds factor limits
C
      DIMENSION AEMIN( MAXMEM ),  AEMAX( MAXMEM )
      DOUBLE PRECISION VBMIN, VBMAX
      LOGICAL INDMIN, INDMAX
C
C  Key to Limited Displacements;
C  Number of Displacement Constraints;
C  Deflection constraints:  maximum deflection for all nodes or
C   magnitude, direction, and node number for each node's constraint
C
      INTEGER LMTDSP, NDSPCN
      DIMENSION DEFMAX( 3 ),
     +          DEFMAG( MAXDEF ), IDRDEF( MAXDEF ), NNDDEF( MAXDEF )
C
C  Frequency limits (negative for lower bound);
C  Number of Frequencies Constrained, Mode number of Constrained freq.
C
      DIMENSION FRQLMT( MAXFQL )
      INTEGER NFRQCN, MODECN( MAXFQL )
C
C  Flag for Rayleight Quotient Frequency Constraint Approximation;
C  Flag for inverting form of Frequency constraint.
C
      LOGICAL FRQAPX, FRQINV
C
C  Structural to total mass modal energy ratios
C
      DIMENSION GAMMAJ( MAXFQL )
      SAVE /OPTIM2/
      COMMON /OPTIM2/ FRQLMT, GAMMAJ, DEFMAX, DEFMAG, IDRDEF, NNDDEF,
     +                AEMIN, AEMAX, VBMIN, VBMAX, INDMIN, INDMAX,
     +                LMTDSP, NDSPCN, NFRQCN, FRQAPX, FRQINV, MODECN,
     +                LMTSTR, NSTRCN, NDUMMY
C-----------------------------------------------------------------------
```

```
C
C  Von Mises Effective Stress Ratio for each element
C
      DIMENSION VMEFSR( MAXCON, MAXLC )
C
C  Strain energies for each element & axial stress values
C
      DIMENSION ENRG( MAXCON+1, MAXLC ), SX(MAXCON)
C
      COMMON /OPTIM3/ VMEFSR, ENRG, SX
      SAVE   /OPTIM3/
C-----------------------------------------------------------------------
C
C Allowable stress values
      DIMENSION ALS(3)
      SAVE   /OPTIM12/
      COMMON /OPTIM12/ ALS
C-----------------------------------------------------------------------
      DIMENSION A(N1,N3),A1(N6,N7),AS(N1,N3),AD(N1,N3),XOBJ(N1),VLB(N1),
     +VUB(N1),G(N2),SCAL(N1),S(N1),G1(N2),G2(N2),B(N3,N3),C(N4), DF(N1),
     +ISC(N2),IC(N3),MS1(N5), ITYPG(N2),IHAC(MAXCON+3),KMAT(K1,K1)
      REAL OBJOLD
C Override selected CONMIN default parameters
      DELFUN = 0.0001
      DABFUN = 0.01
      CTMIN = .0005
      CTLMIN = .001
      CT = -.003
      CTL = -.01
      ITRM = 3
      NFDG = 1
      NSCAL = 0
      LINOBJ = 1
      ITMAX = 75
      NSIDE = 20
      IGOTO = 0
      INDEX=6
      MM=3
C Read NASTRAN data deck to get structural data
      CALL INPUT(SETFUN, NDV, NCON)
C Calculate initial design variable and objective function values
      CALL INIDV(XOBJ, DF)
      IF (NDSPCN .GT. 0 ) NFI=NFI+JOINTS
      IF (NSTRCN .GT. 0 ) NFI=NFI+MEMBS
      DO I = 1,NCON
        ISC(I) = 0
      ENDDO
      DO I = 1,NDV
        VLB(I)=XL
        VUB(I)=XU
      ENDDO
      WRITE(75,*)1
C Calculate objective function value
      CALL CALOBJ(OBJ,DF,XOBJ,NDV,.FALSE.)
C Calculate constraint values
      CALL CALCON(XOBJ,G,ITYPG)
      NAC = 0
      SF=1.0
      PRINT*,'Constraint values'
      DO J=1,NCON
```

```
          PRINT*,'g(j)=',G(J)
          ic(j)=0
          IF (G(J) .GE. CT) THEN
            NAC = NAC + 1
            ic(nac)=j
          ENDIF
        ENDDO
        PRINT*,'Number of active constraints:',NAC
C CALCULATE CONSTRAINT GRADIENTS
        CALL VICKY1(KMAT)
        IF (NDSPCN .GT. 0) THEN
          CALL VICKY2(KMAT,INDEX,AD)
          DO I=1,N7
            DO J=1,N6
              A(J,I)=AD(J,I)
            ENDDO
          ENDDO
          IF (NSTRCN .GT. 0) THEN
            CALL VICKY4(KMAT,INDEX,AS)
            DO K=1,NSTRCN
              DO J=1,N6
                A(J,K+NDSPCN)=AS(J,K)
                ENDDO
            ENDDO
          ENDIF
        ELSE IF (NSTRCN .GT. 0) THEN
        CALL VICKY4(KMAT,INDEX,AS)
        DO I=1,N7
          DO J=1,N6
            A(J,I)=AS(J,I)
          ENDDO
        ENDDO
        ELSE
        PRINT*,'ERROR - NO CONSTRAINTS IDENTIFIED'
        STOP
        ENDIF
        PRINT*,'Constraint Gradients'
        do i=1,n7
          do j=1,n6
          WRITE(6,70)(a(j,i))
          enddo
        enddo
70      FORMAT(6E15.6)
        CALL APXCMN(XOBJ, VLB, VUB, G, A, NDV, NCON, OBJ, DF, IHAC,
       +            RTCNV, INVFLG, MAXCON, MAXNDV, IACT, IVIOL, ITYPG,NVC)
        IF (NVC .EQ. 0) THEN
          READ(85,*)OBJOLD
          IF (ABS((OBJOLD-OBJ)/OBJOLD) .LE. 0.001) THEN
            REWIND(75)
            WRITE(75,*)0
            PRINT*,'COSOPT HAS CONVERGED'
            ENDIF
        ENDIF
        REWIND(85)
        WRITE(85,*)OBJ
        PRINT*,'XOBJ=',(XOBJ(I),I=1,NDV)
        CALL UPDATE(XOBJ,MAXNDV)
        WRITE(6,187)OBJ
  187   FORMAT(5X,21HOBJECTIVE FUNCTION = ,E15.8)
        STOP
        END
```

THE ROLE OF NASTRAN IN THE PRELIMINARY DESIGN CYCLE

H. R. Grooms and V. J. Baipsys
Rockwell International Corporation
Downey, California

SUMMARY

This paper explains how NASTRAN can be utilized advantageously in the preliminary design cycle. The initial portion of the preliminary design process lends itself to programs that can produce multiple configurations or variations on a particular design with minimal cost or effort. The latter portion of the process encompasses refining the design and adding more detailed analyses (particularly for other disciplines). A method for quickly generating balanced spacecraft loading conditions for use in preliminary design and analysis also is explained.

The following additional sections are included:

1. Background
2. Symbols
3. Analytical Process
4. Aerodynamic Load Distributions
5. NASTRAN Applications
6. Conclusion
7. References

BACKGROUND

The preliminary design cycle seeks to obtain general as well as specific information rapidly and inexpensively, yet accurately. The preliminary design cycle (see fig. 1) for spacecraft or space systems usually involves evaluating multiple designs for a given configuration or evaluating several competing configurations. A process for the analysis and evaluation work has been established (ref. 1) and used (ref. 2) for several investigations. This process (fig. 1) starts with a solid representation of the design and evolves into a finite element representation for static and dynamic analysis. Various systems are available for performing the finite element analysis. Two such systems are IDEAS and NASTRAN. The process of preliminary design has, among other things, two objectives that can be opposing: (1) to provide an analytical representation that can be easily revised, and (2) to provide an analytical representation that can be refined as part of the design improvement after a configuration has been accepted. The IDEAS system readily lends itself to objective number (1), while NASTRAN is particularly useful for objective number (2).

Various researchers have suggested approaches (ref. 3 and 4) for optimizing a structural design. The optimization researchers usually start with a given configuration and loading condition. The preliminary design issues addressed in this paper allow consideration for a broader viewpoint. This broader viewpoint asks the following questions:

1. What is a good configuration?
2. What vehicle loads go with a particular configuration?

SYMBOLS

The following symbols are used in this paper:

ρ	air density (slugs/ft^3)
α	angle of attack (rad)
CAD	computer–aided design
q	dynamic pressure (lb/ft^2)
δ	gimbal angle (deg)
V_{gust}	gust velocity (ft/sec)
C_{N_α}	normal force coefficient slope (1/deg)
PL	payload
SRM	solid rocket motor
S_{ref}	surface reference area (ft^2)
T	thrust (lb)
$V_{vehicle}$	vehicle velocity (ft/sec)
V_{wind}	wind velocity (ft/sec)
X_{cg}	X coordinate of the center of gravity (in.)
X_{cp}	X coordinate of the center of pressure (in.)
X_{gimbal}	X coordinate of thrust vector application point (in.)

ANALYTICAL PROCESS

The process starts with a candidate design or configuration that needs to be evaluated. A computer-aided design (CAD) representation is created and serves as a basis for the finite element model. The basic finite element model can serve as the starting point for investigating alternate configurations. It usually takes at least one iteration through a segment (see fig. 2) of the process to get a reasonable estimate of the structural sizing and weights. The first pass-through also provides a good test of the model fineness. The analyst would like the finite element model to be fine enough to give believable stress and deflection predictions; however, it should be crude enough to keep computing costs and time at a low level.

The box entitled "Finite Element Model" (see fig. 2) could utilize any one of a number of different programs. The two most attractive systems for this project were IDEAS and NASTRAN. Table I gives a comparison between the two systems. In order to generate a good preliminary design, both programs (or other comparable ones) should be used: IDEAS (to compare configurations and to select one) and NASTRAN (to provide the starting point for detailed design and certain specialized analyses [e.g., flight control, flexible body loads, etc.]). This is shown in fig. 3.

Considerable effort has been spent in computing vehicle load conditions that are configuration dependent. Any preliminary design can only be as good as the vehicle loads being used. The issue of balanced load conditions is important because in the early stages (preliminary design) of a design meaningful loads are very difficult to obtain. Balanced load conditions on a vehicle allow an analyst to look at the computed stresses and deflections and not be concerned about how the results have been skewed by assumed boundary conditions or unbalanced loads. A balanced load condition is one where the sum of all forces and moments (aerodynamic, inertial, and thrust) acting on the vehicle are zero.

Table I. Comparison of IDEAS and NASTRAN

FINITE ELEMENT MODEL	
IDEAS	**NASTRAN**
Advantages	**Advantages**
• Quick turnaround • Complete solutions • Low cost • Static and dynamic results • Color graphics • Database capability	• Easy to interface with other programs/disciplines • Wide usage in U.S. • Highly portable • Sophisticated solutions available
Disadvantages	**Disadvantages**
• Limited capability to interface with other programs/disciplines • Limited usage in U.S. • Available on limited platforms • Specialty (e.g., buckling, etc.) solutions not available	• Not easy to generate multiple configurations • No built-in color graphics • No convenient database features • "Not so quick" turnaround • Not particularly low cost

AERODYNAMIC LOAD DISTRIBUTIONS

An auxiliary program was set up to provide flexible and rapid inputs to the finite element model for in-flight aerodynamic load distributions on launch vehicles. This method is particularly useful for evaluating the sensitivity of aerodynamic loads due to uncertainties. These uncertainties may be in trajectory parameters, such as: dynamic pressure (q), angle of attack (α), or vehicle center of gravity location (X_{cg}).

Aerodynamic forces normal to the vehicle longitudinal axis cause local loads and bending moments on the vehicle structure. They also require the rocket engines to be deflected (gimbaled) to balance the aerodynamically induced overturning moment on the vehicle. As shown in fig. 4, the loads analysis uses inputs that define certain basic aerodynamic, vehicle, and trajectory parameters.

Aerodynamic inputs consist of the normal aerodynamic force characteristics (transverse to the vehicle longitudinal axis). The aerodynamic normal forces and moments depend on the size and shape of the vehicle elements, and the trajectory parameters including: flight Mach number, angle of attack (α), and dynamic pressure (q). The vehicle size and shape determine the magnitude and shape of the normal force and the location of the airload center of pressure. The normal force is typically represented by distributed normal force coefficient slope, C_{N_α}, along the vehicle. C_{N_α} distributions are obtained empirically or from test data available for similar configurations. Empirical methods (ref. 5) were used for estimating the C_{N_α} variations along vehicle components of various shapes and for a wide range of flight Mach numbers.

The magnitude of α is typically obtained from dynamic trajectory simulations with superimposed wind shear and gusts. If trajectory simulations are not available, an approximate value for α can be estimated by superimposing the wind and gust speeds (ref. 6) on the vehicle speed.

$$\alpha = \mathrm{Tan}^{-1}\left(\frac{V_{wind} + V_{gust}}{V_{vehicle}}\right), \mathrm{rad} \qquad (1)$$

With the trajectory parameters of q and α, and with the C_{N_α} distribution defined along the vehicle, the auxiliary program is used (fig. 5). The method computes the distributed normal forces, net pressures, and the summed forces and moments about the vehicle's center of gravity. Using this method, a vehicle segment of incremental length is subjected to an aerodynamic normal force where the magnitude depends on C_{N_α}, q, and α.

$$\Delta\text{Normal Force} = q\, S_{ref}\, C_{N_\alpha}\, \alpha, \text{ lb/in.} \tag{2}$$

where:

$C_{N\alpha}$ = distributed normal force coefficient slope, 1/(in.-rad)

q = dynamic pressure, $1/2\ \rho\ (V_{vehicle})^2$ (lb/ft^2)

S_{ref} = reference area (ft^2)

α = angle of attack (rad)

ρ = atmospheric density ($slugs/ft^2$)

The above equations are used to compute the normal load distribution along the vehicle. It is then integrated within the auxiliary program to compute the load and moment summations about the center of gravity. The presence of additional elements, such as solid rocket motors (SRMs), can be accounted for by adding their point-load contributions to the total forces and moments.

Static balance calculations are included in the program to determine the amount of engine gimbal angle (δ) required to overcome (or balance) the aerodynamic moment. This is computed from the moment balance between the aerodynamic forces and the engine thrust, as shown below.

$$T\text{Sin}(\delta)(X_{gimbal} - X_{cg}) = \Sigma(C_{N_\alpha})q\alpha S_{ref}(X_{cg} - X_{cp}) \tag{3}$$

The above equation is then solved for the gimbal angle, δ.

$$\delta = \text{Sin}^{-1}\left(\frac{\Sigma(C_{N_\alpha})q\alpha S_{ref}(X_{cg} - X_{cp})}{T(X_{gimbal} - X_{cg})}\right), \text{deg} \tag{4}$$

where:

$\Sigma(C_{N_\alpha})$ = integrated normal force coefficient slope on vehicle (rad)

q = dynamic pressure (lb/ft^2)

S_{ref} = reference area (ft^2)

T = engine thrust (lb)

X_{cg} = center of gravity station (in.)

X_{cp} = center of pressure station (in.)

X_{gimbal} = engine gimbal station (in.)

α = angle of attack (rad)

δ = engine gimbal angle for balancing the aero forces (deg)

For the case when additional engines exist, as in the case of SRMs, the above static moment balance is altered to include such engines. With the SRM and Core subscripts used for the appropriate elements, the moment balance expression becomes:

$$(T_{Core} + T_{SRM})Sin(\delta)(X_{gimbal} - X_{cg}) = q\alpha S_{ref}\left\{C_{N_{\alpha_{Core}}}(X_{cg} - X_{cp_{Core}}) + C_{N_{\alpha_{SRM}}}(X_{cg} - X_{cp_{SRM}})\right\} \quad (5)$$

where $C_{N_{\alpha_{Core}}}$ corresponds to the core stage element and is equivalent to $\Sigma(C_{N_\alpha})$ in the previous moment balance equation.

Then:

$$\delta = Sin^{-1}\left(\frac{q\alpha S_{ref}\left\{C_{N_{\alpha_{Core}}}(X_{cg} - X_{cp_{Core}}) + C_{N_{\alpha_{SRM}}}(X_{cg} - X_{cp_{SRM}})\right\}}{T_{Core}(X_{gimbal} - X_{cg}) + T_{SRM}(X_{gimbal} - X_{cg})}\right), deg \quad (6)$$

With the gimbal angle defined, the axial and tangential thrust values are calculated. These thrust components are then used to compute the axial and tangential accelerations (normal to the vehicle longitudinal axis), which are input into the finite element model.

$$\text{axial acceleration} = \frac{\text{total axial thrust}}{\text{vehicle weight}} \quad (7)$$

$$\text{tangential acceleration} = \frac{\text{total tangential thrust} + \Sigma(\text{normal force})}{\text{vehicle weight}} \quad (8)$$

Key load parameters can be changed easily in the program to see their influence on loads and engine control deflections. A change in dynamic pressure, (q), angle of attack (α), or vehicle center of gravity (X_{cg}) will readily show the sensitivity of aerodynamic loads to such changes.

NASTRAN APPLICATIONS

The Background section of this paper discussed using two different finite element programs for structural analysis. Why not just use one model/program for the entire preliminary design cycle? The two systems, IDEAS and NASTRAN, have different advantages and disadvantages (see table I).

The finite element solver that is internal to IDEAS is a valuable tool, especially when rapid results based on model variations are desired; however, for certain applications, a NASTRAN finite element

representation is much more useful. Figure 6 shows some of the static and dynamic applications that can be supported by the NASTRAN model.

The IDEAS finite element model can be used in its full mass and stiffness representation to compute the first few system mode shapes and natural frequencies of the accepted configuration. This information can be used as a check on the mode shapes and frequencies that are later computed using a reduced dynamic model (e.g., flexible body loads model) generated with NASTRAN.

CONCLUSION

The early portion of the preliminary design cycle makes the use of the finite element code in IDEAS attractive because a vehicle analysis can be quickly redone after sizing changes are made. This paper describes a procedure for preliminary design and shows how NASTRAN can be used as a vital tool in that process. Additionally, a method for setting up balanced vehicle load conditions, as an integral part of that procedure, has been explained in detail. The challenge in the preliminary design cycle is to create a large amount of meaningful information rapidly and inexpensively, to use the preliminary design analytical representation to interact with many disciplines, and to support the evolution of a detailed design.

The later stages of the preliminary design can be effectively handled by NASTRAN because of its ability to:

1. Handle many thousands of degree problems relatively cheaply
2. Run on many different platforms
3. Easily interface with other programs/data sources

REFERENCES

1. Grooms, H. R.; DeBarro, C. F.; and Paydarfar, S.: "What is an Optimal Spacecraft Structure?" April 1990. Proceedings of the 31st AIAA/ASME/ASCE/AHS/ACS Structures, Structural Dynamics, and Materials Conference, Long Beach, California.

2. Grooms, H. R.; Blanchard, W.; Hawthorne, D.; and Fisk K.: "An Automated Database System for Preliminary Spacecraft Design," August 1992. Proceedings of the 1992 ASME International Computers in Engineering Conference and Exposition, San Francisco, California.

3. Skelton, R. E.; Hanks, B. R.; and Smith, M.: "Structure Redesign for Improved Dynamic Response," Journal of Guidance, Control, and Dynamics, vol. 15, no. 5, Sept to Oct, 1992.

4. Raman, A.: "A Non-Iterative Method of Structural Optimization for Static, Dynamic, and Response Problems." Computers and Structures, vol. 34, no. 2, 1990.

5. NASA TN D-3283, "An Empirical Method for Determining Static Distributed Aerodynamic Loads On Axisymmetric Multistage Launch Vehicles." March 1966.

6. NASA TM 82473, "Terrestrial Environment (Climatic) Criteria Guidelines For Use In Aerospace Vehicle Development." Updated June, 1982. (Formerly NASA TM-X-64757).

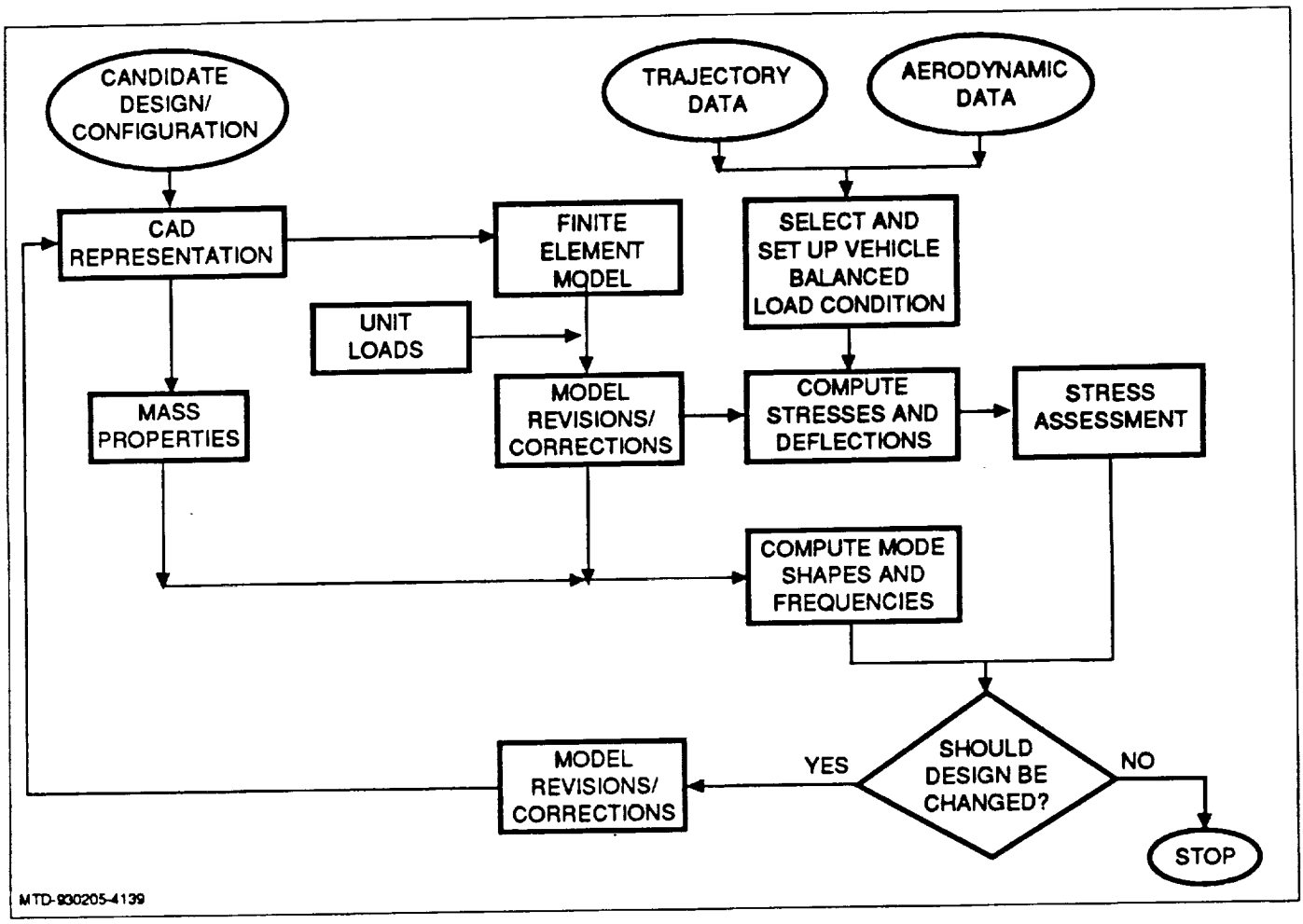


Figure 1. Preliminary Design and Analysis Cycle

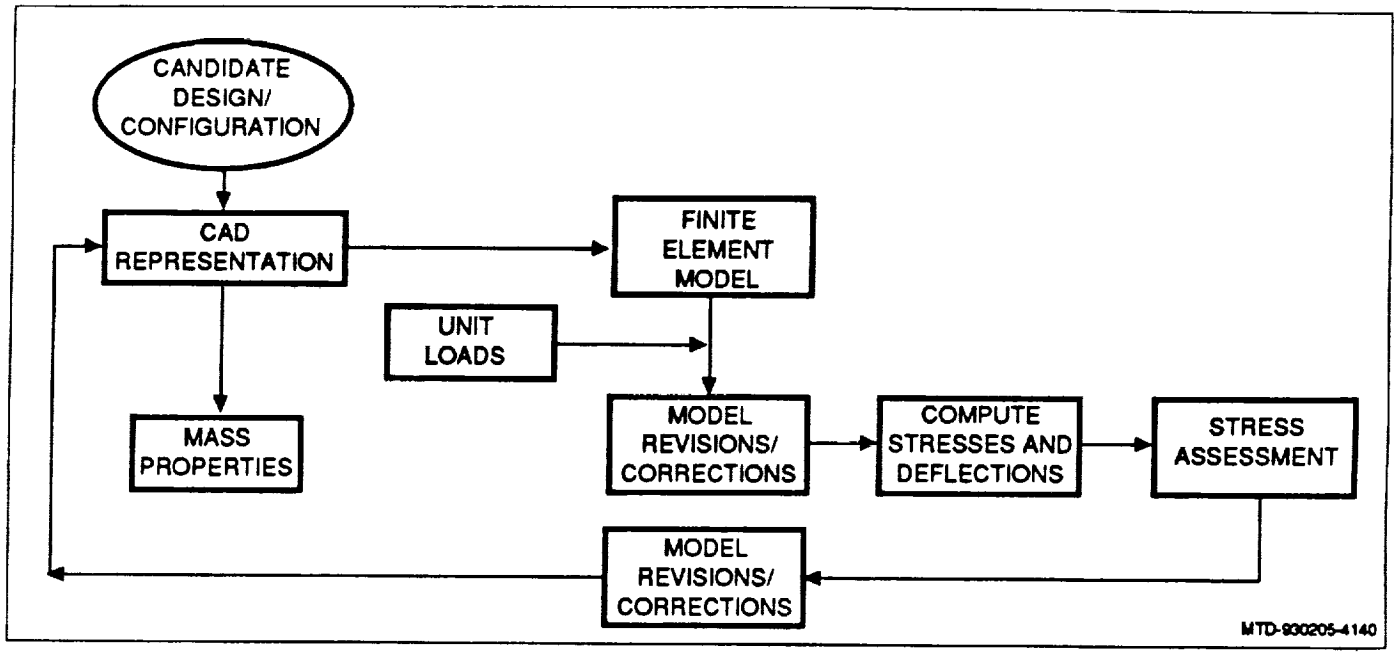


Figure 2. Units Loads are Used to Assess the Initial Design

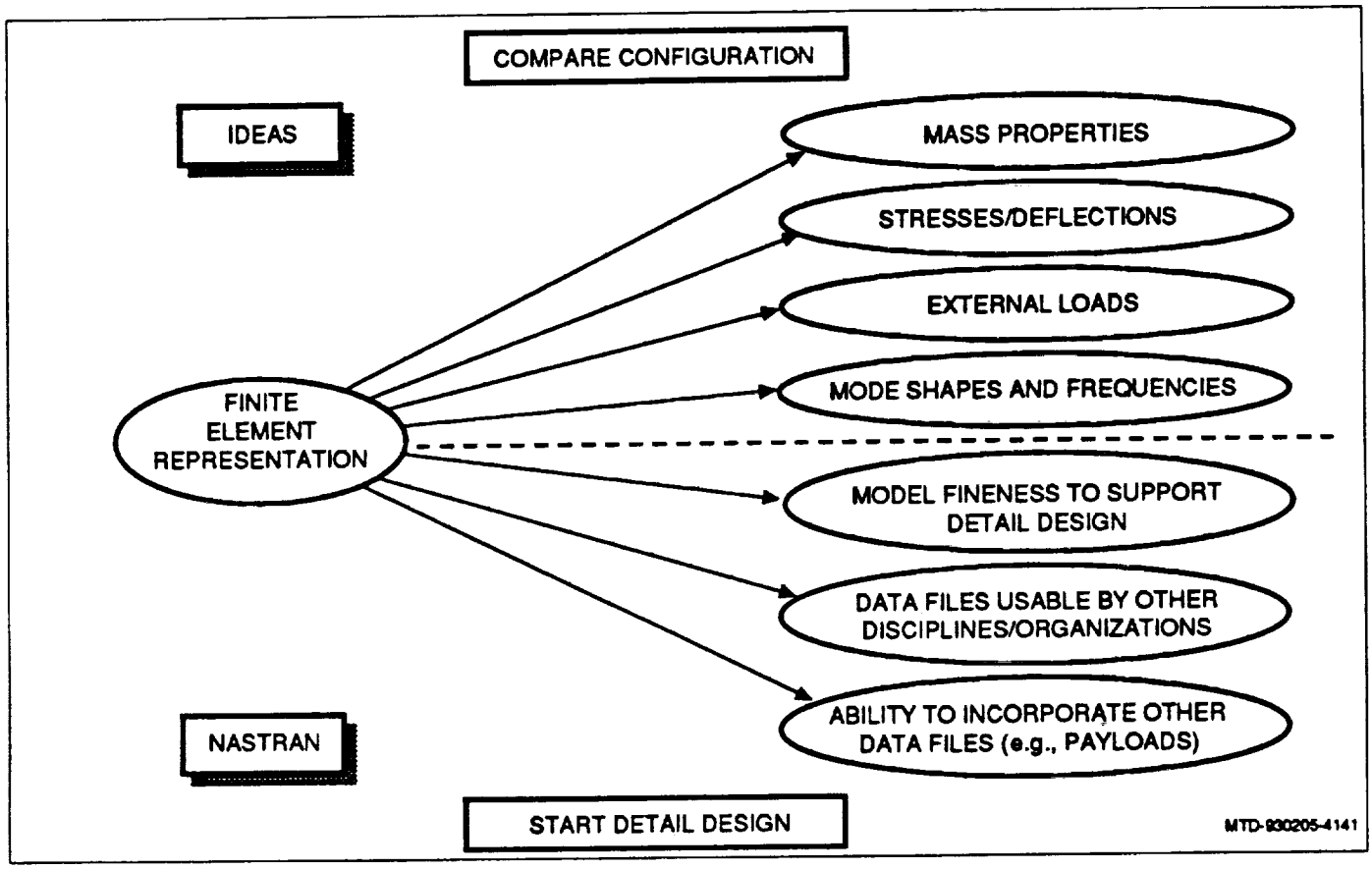


Figure 3. NASTRAN and IDEAS are Used for Different Purposes

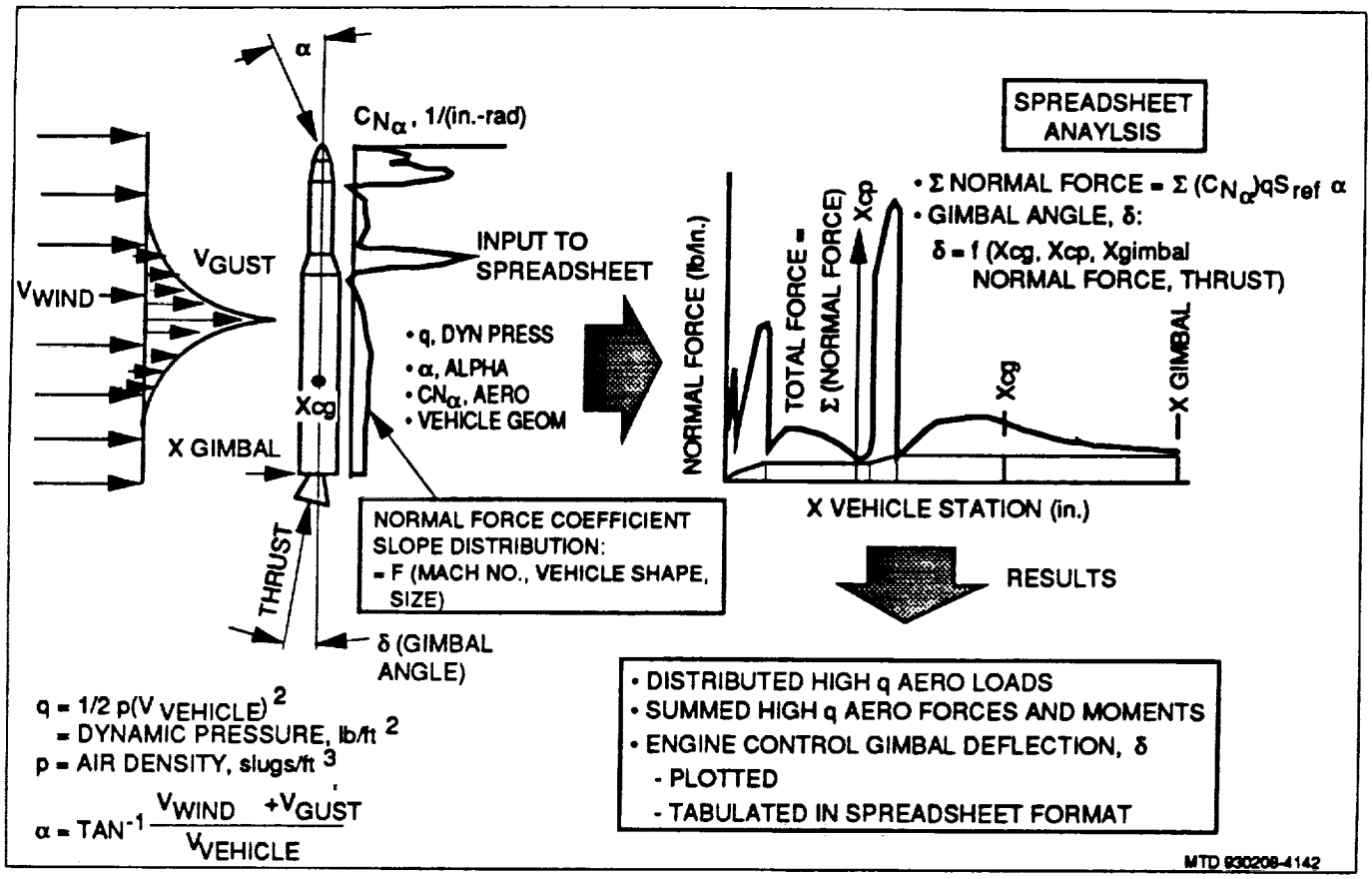


Figure 4. An Example of High Dynamic Pressure Region Vehicle Loads

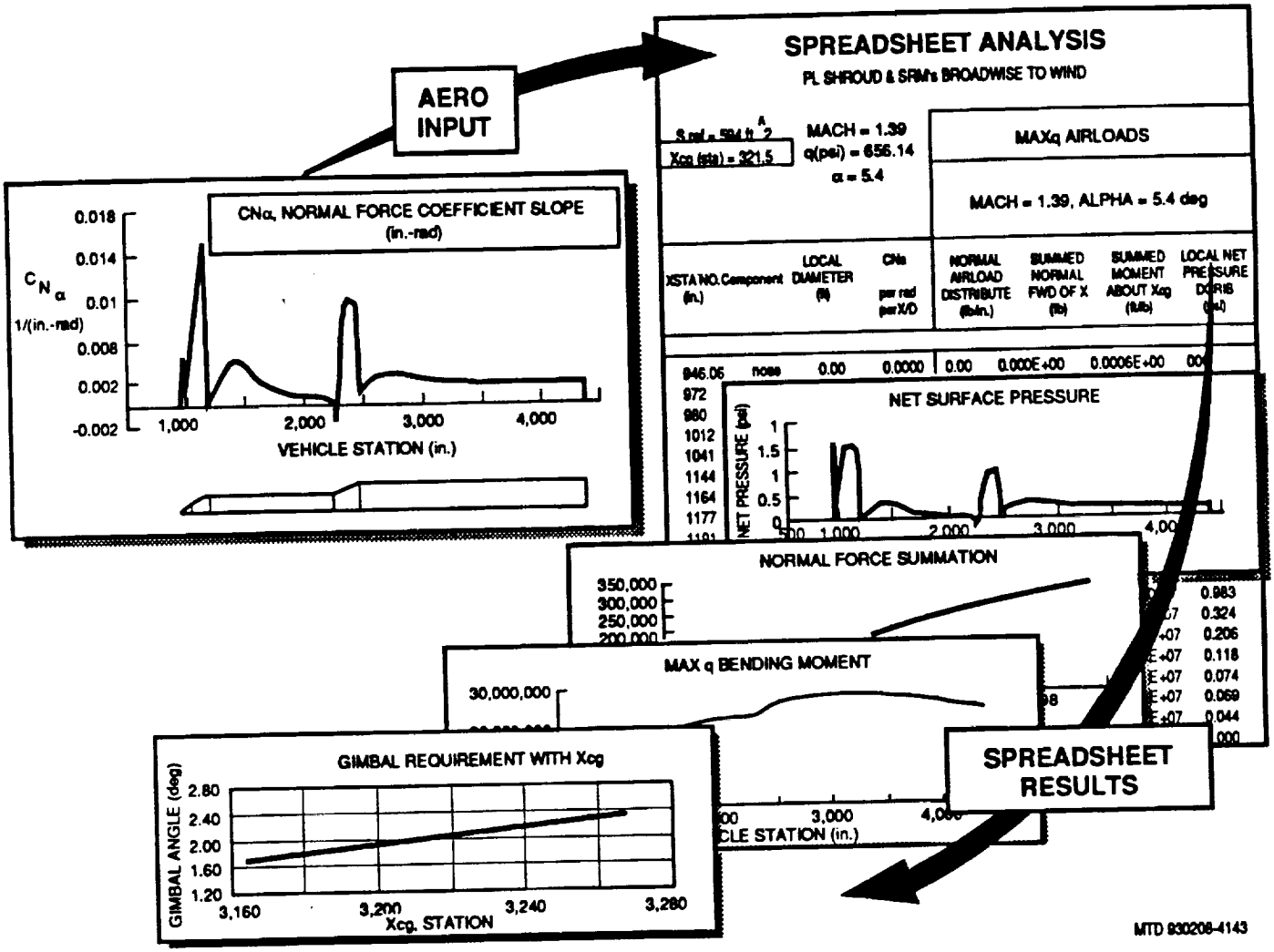


Figure 5. The Aerodynamic Influence is Displayed Different Ways

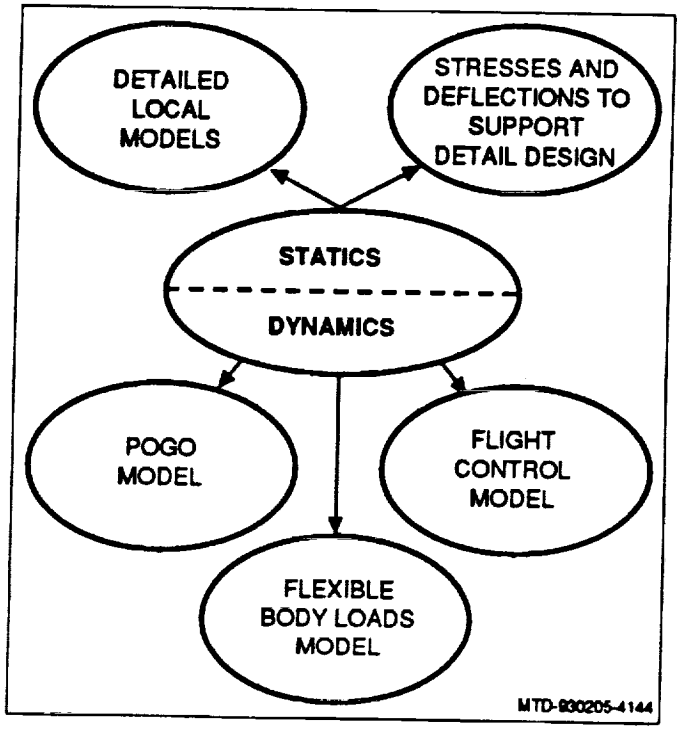


Figure 6. Applications of a NASTRAN Representation

OMIT

APPENDIX

Unpublished Papers Presented at the Twentieth NASTRAN Users' Colloquium

Colorado Springs, CO
April 27 - May 1, 1992

A METHODOLOGY TO MODEL PHYSICAL CONTACT BETWEEN STRUCTURAL COMPONENTS IN NASTRAN

Annappa A. Prabhu
G. E. Government Services, Houston Texas

SUMMARY

Two components of a structure which are located side by side, will come in contact by certain force and will transfer the compressive force along the contact area. If the force acts in the opposite direction, the elements will separate and no force will be transferred. If this contact is modelled, the load path will be correctly represented, and the load redistribution results in more realistic stresses in the structure. This is accomplished by using different sets of rigid elements for different loading conditions, or by creating multipoint constraint sets. Comparison of these two procedures is presented for a 4 panel unit (PU) stowage drawer installed in an experiment rack in the Spacelab Life Sciences (SLS-2) payload.

INTRODUCTION

The Spacelab is a reusable laboratory that is carried in the cargo bay of the Space Shuttle. Experiments in several different disciplines such as astronomy, life sciences, and material science are accommodated in this modular laboratory for various Shuttle missions. The experiment hardware is mounted in the experiment racks located in either side of the module, in overhead lockers, or in the center aisle, as shown in Figure 1.

4PU STOWAGE DRAWER

Configuration

The 4 Panel Unit Stowage Drawer is mounted in the experiment rack used in the SLS-2 Mission. The experiment equipment and the accessories are stowed in the drawer. The finite element model of the drawer, with its coordinate system, is shown in Figure 2. The drawer is connected to the slide with 6 screws on each slide, and the slides are connected to the rack posts. The front panel is latched to the front rack posts. Two configurations of the slides are examined.

Case 1: The contact surface is normal to the X-axis, as shown in Figure 4, which is the actual configuration. The slide shown is schematic, and not the actual slide.

Case 2: The contact surface is inclined. This is achieved by raising the slide by 12.7 mm as shown in Figure 5.

The Method of Modelling the Contact

During liftoff and landing flight events, the Shuttle and its payload are exposed to quasi-static and random loads. The +X force brings the right slide and drawer in contact. As a result, this force is transferred to the slide throughout the length of the slide and not just by the screws. When the force acts to the left (-X), the contact along the length is lost and the right slide is connected by screws only. This time the contact takes place between the left slide and the drawer.

Generally, this is modelled using rigid elements. For load case 101, which includes +X force (see Table 1), all the contacts are modelled between the right slide and the drawer, and the analysis is completed. For the load case 103, which includes -X force, contact will be modelled between the left slide and the drawer with a new set of rigid elements, removing the old set of elements, and a second analysis will be performed. This means post-processing will be performed on two output files. The rigid elements simulating contact are shown in Figure 3. These are included in the analyses, as needed.

Alternately, the contact is modelled with a multipoint constraint equation in place of the rigid element. In this method, a different set of MPC equation can be written for a different subcase, resulting in a single analysis for multiple subcases.

CASE 1. CONTACT SURFACE NORMAL TO THE GLOBAL X-AXIS

Modelling of Contact by MPC Equation

A rigid link is used to write the MPC equation, as shown in Figure 6 (ref 1.). Since the physical contact cannot resist moments, no rotations will be allowed at the end of the rigid link. In the current case, the link is horizontal, i.e., $\Delta L = \Delta X = u_1$.
The MPC equation is $u_{1A}-u_{1B} = 0$. The MPC set 1 is written for the subcase 101 to represent right slide contact, and the MPC set 2 is written for the subcase 103 to represent the left slide contact. The MPC equations and the MPC forces are shown in Table 2. The grid points shown are on the slide. Grid points 471 and 472 show forces in opposite directions, indicating tension and lack of contact. In this situation, these equations should be removed and reanalysis must be performed. In the current analysis, this is not pursued.

Modelling with CRIGD2 Elements

The elements modelled and the results for subcase 101 and subcase 103 are shown on Tables 3 and 4, respectively. The dependent degree of freedom is 1. The equation generated corresponds to row 1 of equation 56 (ref. 2) shown below.

$$\begin{Bmatrix} u_{A_1} \\ u_{A_2} \\ u_{A_3} \\ u_{A_4} \\ u_{A_5} \\ u_{A_6} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & (z_B-z_A) & -(y_B-y_A) \\ 0 & 1 & 0 & -(z_B-z_A) & 0 & (x_B-x_A) \\ 0 & 0 & 1 & (y_B-y_A) & -(x_B-x_A) & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_{B_1} \\ u_{B_2} \\ u_{B_3} \\ u_{B_4} \\ u_{B_5} \\ u_{B_6} \end{Bmatrix} , \qquad (56)$$

In the equation, $(Z_B - Z_A)$ corresponds to Z_{AB} on Figure 6 which is zero for this case, and $(Y_B - Y_A)$ is also zero. Hence, the equations generated are the same as the MPC equations and the results from both the analyses will be identical.

Discussions of the Two Methods

As expected, the results from both of these methods are the same.

CASE 2. CONTACT SURFACE INCLINED TO THE GLOBAL X-AXIS

Only subcase 101, which involves +X and +Z loads, will be used in the following analyses. Due to these forces, contact will be made in the X and Z directions as shown in Figure 5.

Modelling with MPC Equations

MPC equations are written to satisfy the geometry of the rigid links shown in Figure 6. As stated before, no rotation will be allowed.

Hence $\Delta L = \frac{X_{AB}}{L} u_1 + \frac{Z_{AB}}{L} u_3$ (1)

from the geometry of the inclined link in Figure 5,
X_{AB} = 18.606 mm, Z_{AB} = 12.7 mm , L= 22.527 mm
Substituting in equation (1)
$\Delta L = .8259\ u_1 + .5638\ u_3$
$\Delta L_A = .8259\ u_{1A} + .5638\ u_{3A}$
$\Delta L_B = .8259\ u_{1B} + .5638\ u_{3B}$

to satisfy the condition $\Delta L_A - \Delta L_B = 0$, the MPC equation is
$.8259\ (u_{1A} - u_{1B}) + .5638\ (u_{3A} - u_{3B}) = 0$ (2)

Table 5 shows all the MPC equations input for all the contacts, followed by the forces of multipoint constraint, in the grid points on the slides.

Modelling with CRIGD2 Elements

CRIGD2 are modelled with components 1 and 3 as dependent degrees of freedom to simulate the contact in X and Z directions. The constraint equations generated correspond to rows 1 and 3, in the equation 56. The term $(Z_B - Z_A)$ in row 1 and $(X_B - X_A)$ in row 3 are non-zero. These terms correspond to component 5, and it is expected that constraint moment R2 will be generated. The list of elements and the results are tabulated in Table 6.

Modelling with CRIGDR Elements

CRIGDR elements are modelled with component 1 as the dependent degree of freedom. The remaining 5 translational components are considered as reference degrees of freedom (ref. 2). Equation 48 (ref. 2) is used in the element formulation shown below.

$$(u_{A1} - u_{B1})\, l_1 + (u_{A2} - u_{B2})\, l_2 + (u_{A3} - u_{B3})\, l_3 = 0 \qquad (48)$$

In this equation, direction cosine $l_2 = 0$, $l_1 = \frac{XAB}{L}$, $l_3 = \frac{ZAB}{L}$ which essentially is MPC equation (2), and the results from this analysis will be same as from the MPC equation.

A list of the elements and the results are tabulated in Table 7.

Comparison of the Three Analyses

It is shown that the formulation of MPC equations and the CRIGDR are identical, and the results tabulated in Tables 5 and 7 are identical as expected. The CRIGD2 results are different than the other two because this involves rotations. In this instance, R2 moments are generated as expected and the Z components are off by about ±20 percent.

CONCLUSIONS

The best way to model contact is by writing MPC equations since a single analysis, is possible for multiple subcases. CRIGDR is the second choice.

REFERENCES

1. Harry G. Schaeffer: MSC/NASTRAN Primer, Static and Normal Modes Analysis, Schaeffer Analysis Inc, Mont Vernon, 1979, pp.143-145

2. The NASTRAN Theoretical Manual: NASA SP-221(06), National Aeronautics And Space Administration, Washington D.C., 1981

TABLE 1 TOTAL APPLIED FORCE ON THE STRUCTURE (NEWTONS)

DIRECTION	X	Y	Z
SUBCASE 101	1001.2	175.3	276.4
SUBCASE 103	-897.0	175.3	276.4

TABLE 2 PARTIAL INPUT AND RESULTS - CASE 1
SUBCASES 101,103 - USE OF MPC EQUATIONS

```
$ MPC EQUATIONS TO SIMULATE CONTACT IN X-DIR
$ FOR SUBCASE 101 CASE 1
MPC            1      454       1      1.0     1070       1     -1.0
MPC            1      471       1      1.0     1104       1     -1.0
MPC            1      472       1      1.0     1138       1     -1.0
MPC            1      473       1      1.0     1172       1     -1.0
MPC            1      474       1      1.0     1206       1     -1.0
MPC            1      475       1      1.0     1240       1     -1.0
MPC            1      476       1      1.0     1291       1     -1.0
MPC            1      477       1      1.0     1342       1     -1.0
MPC            1      478       1      1.0     1376       1     -1.0
$ MPC EQUATIONS TO SIMULATE CONTACT IN -X-DIR
$ FOR SUBCASE 103 CASE 1
MPC            2     5454       1      1.0     6070       1     -1.0
MPC            2     5471       1      1.0     6104       1     -1.0
MPC            2     5472       1      1.0     6138       1     -1.0
MPC            2     5473       1      1.0     6172       1     -1.0
MPC            2     5474       1      1.0     6206       1     -1.0
MPC            2     5475       1      1.0     6240       1     -1.0
MPC            2     5476       1      1.0     6291       1     -1.0
MPC            2     5477       1      1.0     6342       1     -1.0
MPC            2     5478       1      1.0     6376       1     -1.0
```

F O R C E S O F M U L T I - P O I N T C O N S T R A I N T

SUBCASE 101 CASE 1

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
471	G	2.000663E+00	0.0	0.0	0.0	0.0	0.0
472	G	9.827873E-01	0.0	0.0	0.0	0.0	0.0
473	G	-2.369567E-01	0.0	0.0	0.0	0.0	0.0
474	G	-8.407188E-01	0.0	0.0	0.0	0.0	0.0
475	G	-4.436086E+00	0.0	0.0	0.0	0.0	0.0
476	G	-5.899112E+00	0.0	0.0	0.0	0.0	0.0
477	G	-2.644422E+00	0.0	0.0	0.0	0.0	0.0
478	G	-4.057768E+00	0.0	0.0	0.0	0.0	0.0

SUBCASE 103 CASE 1

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
5471	G	-1.609514E+00	0.0	0.0	0.0	0.0	0.0
5472	G	-7.575473E-01	0.0	0.0	0.0	0.0	0.0
5473	G	3.009112E-01	0.0	0.0	0.0	0.0	0.0
5474	G	8.103580E-01	0.0	0.0	0.0	0.0	0.0
5475	G	3.798277E+00	0.0	0.0	0.0	0.0	0.0
5476	G	4.936887E+00	0.0	0.0	0.0	0.0	0.0
5477	G	2.151909E+00	0.0	0.0	0.0	0.0	0.0
5478	G	3.283999E+00	0.0	0.0	0.0	0.0	0.0

TABLE 3 PARTIAL INPUT AND RESULTS - CASE 1
SUBCASE 101 - USE OF CRIGD2 ELEMENTS

```
$ RIGID ELEMENTS MODELED TO SIMULATE CONTACT IN X-DIR
$ FOR LOADCASE 101   CASE 1
CRIGD2         480      454      1070         1
CRIGD2         481      471      1104         1
CRIGD2         482      472      1138         1
CRIGD2         483      473      1172         1
CRIGD2         484      474      1206         1
CRIGD2         485      475      1240         1
CRIGD2         486      476      1291         1
CRIGD2         487      477      1342         1
CRIGD2         488      478      1376         1
```

SUBCASE 101 CASE1

FORCES OF MULTI-POINT CONSTRAINT

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
471	G	2.000663E+00	0.0	0.0	0.0	0.0	0.0
472	G	9.827873E-01	0.0	0.0	0.0	0.0	0.0
473	G	-2.369567E-01	0.0	0.0	0.0	0.0	0.0
474	G	-8.407188E-01	0.0	0.0	0.0	0.0	0.0
475	G	-4.436086E+00	0.0	0.0	0.0	0.0	0.0
476	G	-5.899112E+00	0.0	0.0	0.0	0.0	0.0
477	G	-2.644422E+00	0.0	0.0	0.0	0.0	0.0
478	G	-4.057767E+00	0.0	0.0	0.0	0.0	0.0

TABLE 4 PARTIAL INPUT AND RESULTS -CASE 1
SUBCASE 103 - USE OF CRIGD2 ELEMENTS

```
$ RIGID ELEMENTS MODELED TO SIMULATE CONTACT IN -X-DIR
$ FOR SUBCASECASE 103   CASE 1
CRIGD2         5480     5454     6070         1
CRIGD2         5481     5471     6104         1
CRIGD2         5482     5472     6138         1
CRIGD2         5483     5473     6172         1
CRIGD2         5484     5474     6206         1
CRIGD2         5485     5475     6240         1
CRIGD2         5486     5476     6291         1
CRIGD2         5487     5477     6342         1
CRIGD2         5488     5478     6376         1
```

FORCES OF MULTI-POINT CONSTRAINT

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
5471	G	-1.609514E+00	0.0	0.0	0.0	0.0	0.0
5472	G	-7.575473E-01	0.0	0.0	0.0	0.0	0.0
5473	G	3.009112E-01	0.0	0.0	0.0	0.0	0.0
5474	G	8.103580E-01	0.0	0.0	0.0	0.0	0.0
5475	G	3.798277E+00	0.0	0.0	0.0	0.0	0.0
5476	G	4.936887E+00	0.0	0.0	0.0	0.0	0.0
5477	G	2.151909E+00	0.0	0.0	0.0	0.0	0.0
5478	G	3.283999E+00	0.0	0.0	0.0	0.0	0.0

TABLE 5 PARTIAL INPUT AND RESULTS -CASE 2
SUBCASE 101 - USE OF MPC EQUATIONS

```
$ MPC EQUATIONS TO SIMULATE CONTACT IN X AND Z DIR
$ FOR SUBCASE 101 CASE 2 (INCLINED SURFACE)
MPC             1      454       1  0.8259     1070       1 -0.8259        +MPC1
+MPC1                  454       3  0.5638     1070       3 -0.5638
MPC             1      471       1  0.8259     1104       1 -0.8259        +MPC2
+MPC2                  471       3  0.5638     1104       3 -0.5638
MPC             1      472       1  0.8259     1138       1 -0.8259        +MPC3
+MPC3                  472       3  0.5638     1138       3 -0.5638
MPC             1      473       1  0.8259     1172       1 -0.8259        +MPC4
+MPC4                  473       3  0.5638     1172       3 -0.5638
MPC             1      474       1  0.8259     1206       1 -0.8259        +MPC5
+MPC5                  474       3  0.5638     1206       3 -0.5638
MPC             1      475       1  0.8259     1240       1 -0.8259        +MPC6
+MPC6                  475       3  0.5638     1240       3 -0.5638
MPC             1      476       1  0.8259     1291       1 -0.8259        +MPC7
+MPC7                  476       3  0.5638     1291       3 -0.5638
MPC             1      477       1  0.8259     1342       1 -0.8259        +MPC8
+MPC8                  477       3  0.5638     1342       3 -0.5638
MPC             1      478       1  0.8259     1376       1 -0.8259        +MPC9
+MPC9                  478       3  0.5638     1376       3 -0.5638
```

SUBCASE 101 CASE 2 (INCLINED SURFACE)

F O R C E S O F M U L T I - P O I N T C O N S T R A I N T

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
471	G	2.349882E+00	0.0	1.604145E+00	0.0	0.0	0.0
472	G	1.092854E+00	0.0	7.460357E-01	0.0	0.0	0.0
473	G	1.307303E+00	0.0	8.924297E-01	0.0	0.0	0.0
474	G	-1.051737E+00	0.0	-7.179675E-01	0.0	0.0	0.0
475	G	-4.542925E+00	0.0	-3.101224E+00	0.0	0.0	0.0
476	G	-6.415951E+00	0.0	-4.379844E+00	0.0	0.0	0.0
477	G	-2.202427E+00	0.0	-1.503485E+00	0.0	0.0	0.0
478	G	-3.147465E+00	0.0	-2.148615E+00	0.0	0.0	0.0

TABLE 6 PARTIAL INPUT AND RESULTS - CASE 2
SUBCASE 101 - USE OF CRIGD2 ELEMENTS

```
$ RIGID ELEMENTS MODELED TO  CONTACT IN X AND Z-DIR
$ DUE SUBCASE 101 CASE 2
CRIGD2         480      454     1070       13
CRIGD2         481      471     1104       13
CRIGD2         482      472     1138       13
CRIGD2         483      473     1172       13
CRIGD2         484      474     1206       13
CRIGD2         485      475     1240       13
CRIGD2         486      476     1291       13
CRIGD2         487      477     1342       13
CRIGD2         488      478     1376       13
SUBCASE 101              CASE 2 (INCLINED SURFACE)
F O R C E S   O F   M U L T I - P O I N T   C O N S T R A I N T
```

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
471	G	2.494523E+00	0.0	2.121828E+00	0.0	3.069774E-01	0.0
472	G	1.101543E+00	0.0	9.281644E-01	0.0	1.291087E-01	0.0
473	G	1.320337E+00	0.0	5.013174E-01	0.0	-2.929534E-01	0.0
474	G	-1.060023E+00	0.0	-5.814678E-01	0.0	1.040862E-01	0.0
475	G	-4.534455E+00	0.0	-2.838911E+00	0.0	1.877255E-01	0.0
476	G	-6.437643E+00	0.0	-3.742413E+00	0.0	4.775038E-01	0.0
477	G	-2.220512E+00	0.0	-1.282993E+00	0.0	1.704635E-01	0.0
478	G	-3.182428E+00	0.0	-1.670981E+00	0.0	3.672205E-01	0.0

TABLE 7 PARTIAL INPUT AND RESULTS CASE 2
SUBCASE 101 -USE OF CRIGDR ELEMENTS

```
$ RIGID ELEMENTS TO SIMULATE CONTACT IN X AND Z -DIR
$ FOR  SUBCASE 101 CASE 2 (INCLINED SURFACE)
CRIGDR         480      454     1070        1
CRIGDR         481      471     1104        1
CRIGDR         482      472     1138        1
CRIGDR         483      473     1172        1
CRIGDR         484      474     1206        1
CRIGDR         485      475     1240        1
CRIGDR         486      476     1291        1
CRIGDR         487      477     1342        1
CRIGDR         488      478     1376        1
SUBCASE 101              CASE 2 (INCLINED PLANE)
F O R C E S   O F   M U L T I - P O I N T   C O N S T R A I N T
```

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
471	G	2.378163E+00	0.0	1.623320E+00	0.0	0.0	0.0
472	G	1.119243E+00	0.0	7.639883E-01	0.0	0.0	0.0
473	G	1.316421E+00	0.0	8.985810E-01	0.0	0.0	0.0
474	G	-1.053347E+00	0.0	-7.190084E-01	0.0	0.0	0.0
475	G	-4.543006E+00	0.0	-3.101028E+00	0.0	0.0	0.0
476	G	-6.414644E+00	0.0	-4.378596E+00	0.0	0.0	0.0
477	G	-2.183337E+00	0.0	-1.490333E+00	0.0	0.0	0.0
478	G	-3.126624E+00	0.0	-2.134214E+00	0.0	0.0	0.0

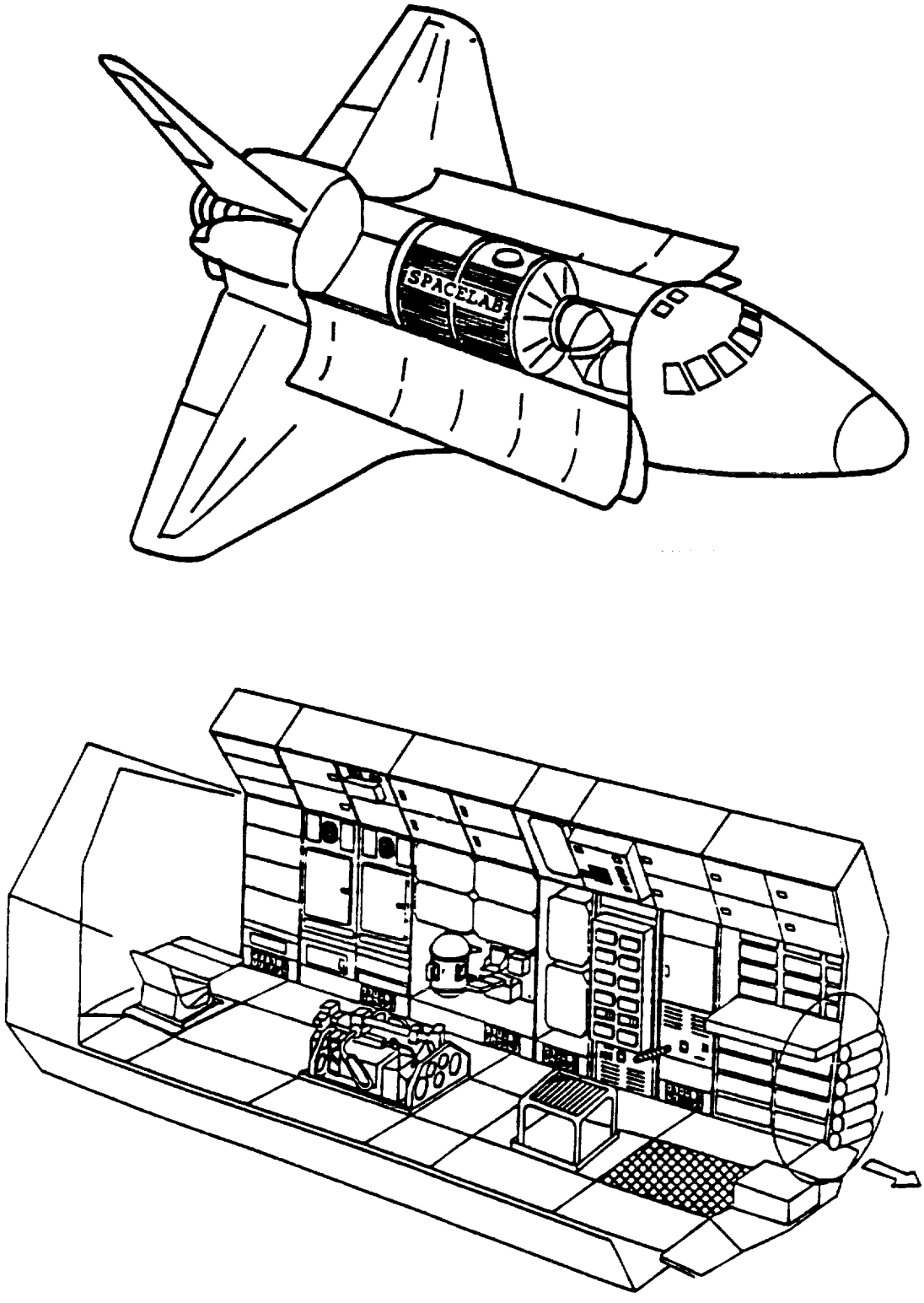


FIG. 1. TYPICAL SPACELAB CONFIGURATION

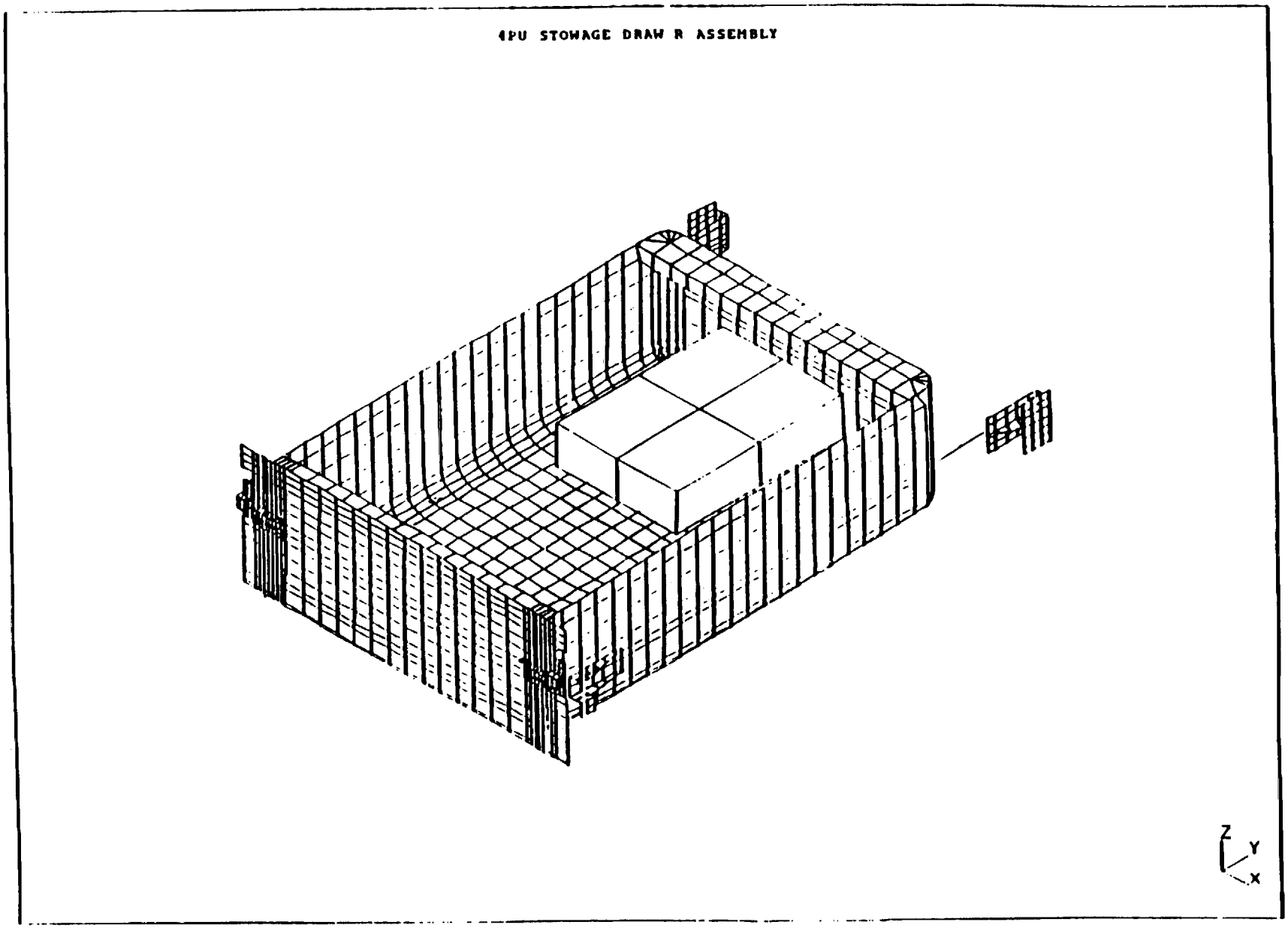


FIG. 2 FINITE ELEMENT MODEL

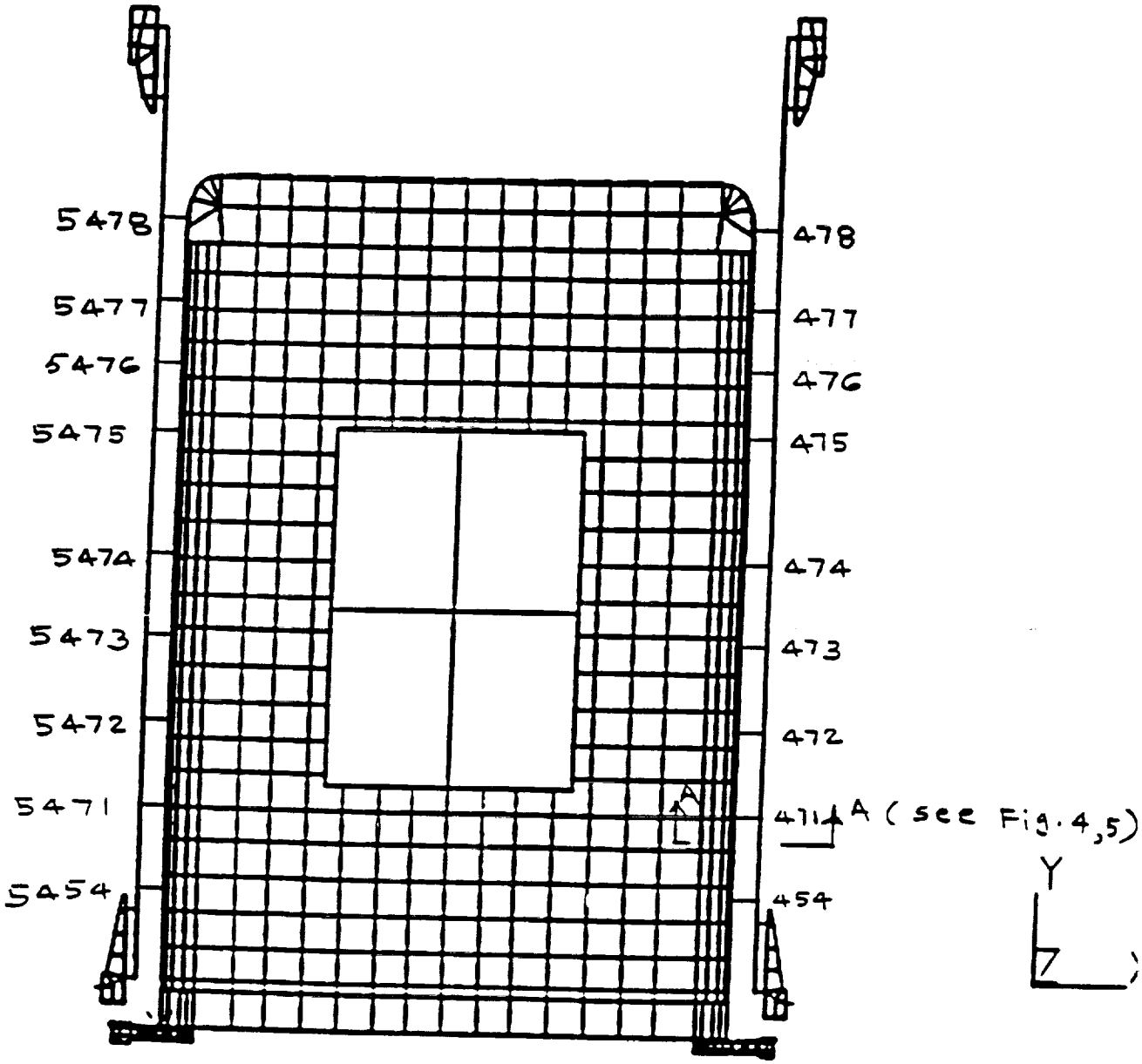


FIG 3. PLAN VIEW SHOWING RIGID ELEMENTS
CONNECTING DRAWER TO SLIDE
(The numbers shown are grid points on the slide)

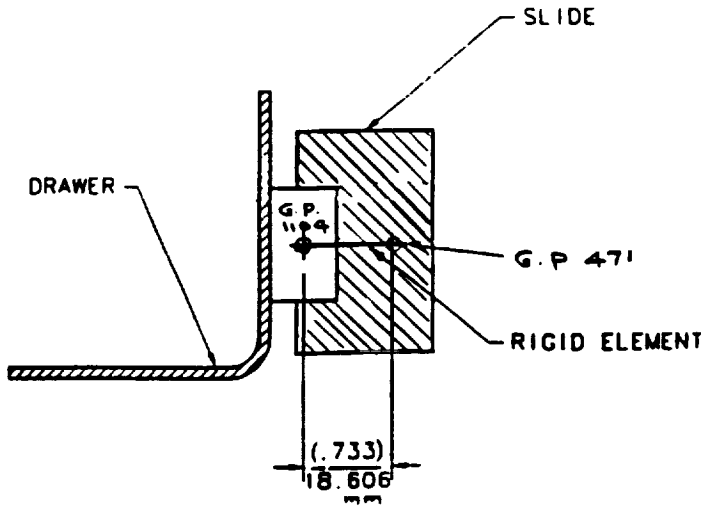


FIG. 4 SECTION A-A FOR CASE 1

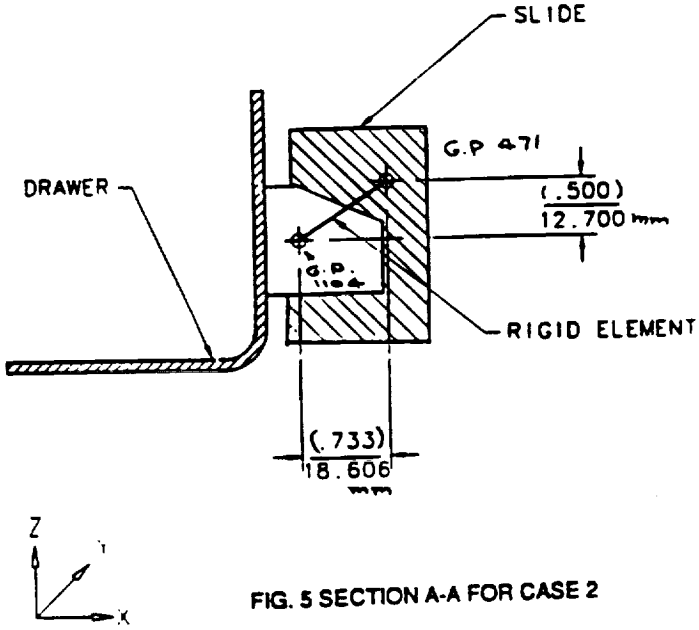


FIG. 5 SECTION A-A FOR CASE 2

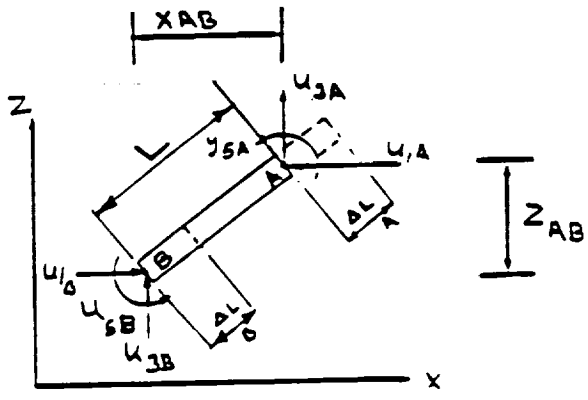


FIG. 6 PLANAR RIGID LINK

30-39
190580
p. 10

BENCHMARKING THE QUAD4/TRIA3 ELEMENT

BY

STEPHEN M. PITROF & VIPPERLA B. VENKAYYA

WRIGHT LABORATORY
WRIGHT-PATTERSON AFB OHIO

N94-17837

INTRODUCTION

The QUAD4 and TRIA3 elements are the primary plate/shell elements in NASTRAN*. These elements enable the user to analyze thin plate/shell structures for membrane, bending and shear phenomena. They are also very new elements in the NASTRAN library. These elements are extremely versatile and constitute a substantially enhanced analysis capability in NASTRAN. However, with the versatility comes the burden of understanding a myriad of modeling implications and their effect on accuracy and analysis quality. The validity of many aspects of these elements were established through a series of benchmark problem results and comparison with those available in the literature and obtained from other programs like MSC/NASTRAN[2] and CSAR/NASTRAN[3]. Nevertheless such a comparison is never complete because of the new and creative use of these elements in complex modeling situations. One of the important features of QUAD4 and TRIA3 elements is the offset capability which allows the midsurface of the plate to be noncoincident with the surface of the grid points. None of the previous elements, with the exception of bar (beam), has this capability. The offset capability played a crucial role in the design of QUAD4 and TRIA3 elements. It allowed modeling layered composites, laminated plates and sandwich plates with the metal and composite face sheets. Even though the basic implementation of the offset capability is found to be sound in the previous applications, there is some uncertainty in relatively simple applications. The main purpose of this paper is to test the integrity of the offset capability and provide guidelines for its effective use. For the purpose of simplicity, references in this paper to the QUAD4 element will also include the TRIA3 element.

BACKGROUND

The QUAD4 element was added to the COSMIC/NASTRAN element library in 1987. Although similar in use to the MSC/NASTRAN QUAD4 element of 1980, there are differences in the theoretical formulation of the two. These differences are primarily in the hardening of shear deformation and numerical integration.

The formulation for the QUAD4 isoparametric quadrilateral element incorporates a bilinear variation of geometry and deformation within the element. The QUAD4 element has 5 degrees of freedom (dof) per node, i.e., the stiffness for rotation about the normal to the mid-surface at each node is not defined. Furthermore, it is assumed that plane sections remain plane and that the variation of strains through the thickness is linear. In addition, direct strain through the thickness is neglected (assumed to be zero).

The QUAD4 element may be used to model either membrane or bending behavior, or both. Transverse shear flexibility may be requested as well as the coupling of membrane and bending behaviors using nodal offsets or linear variation of material properties through the thickness. In addition, the QUAD4 element is capable of representing laminated composite materials, with an option to compute interlaminar shear stresses and layer failure indices.

*NASTRAN without qualification refers to COSMIC/NASTRAN[1].

The transverse shear stiffness is numerically conditioned to enhance the accuracy of the element for a wide range of modeling practices including very thick or thin elements, high aspect ratio elements, and skewed elements.[4]

FEATURES OF THE QUAD4

The QUAD4 element gives the NASTRAN user an accurate, all-purpose plate/shell/membrane element. It can be used in place of all QUAD and QDMEM elements. The QUAD4 element uses a linear, isoparametric formulation with bilinear variation of geometry and deformation. It can be used to model the following types of plates:

- Membrane plates
- Bending plates
  - Membrane/bending (without nonlinear coupling)
  - Membrane/bending (with offset coupling)
- Plates offset from the grid point plane
- Layered composite plates
- Laminated plates
- Sandwich plates (metal and composite face sheets)
- Thin and Thick plates

USE OF THE OFFSET CAPABILITY AND ITS IMPLICATIONS

There are several different ways to specify plate offsets in NASTRAN. They are as follows:

- Z0 field on CQUAD4 bulk data card
- Z0 field on PSHELL bulk data card
- Z0 field on PCOMP bulk data card
- Use of rigid element (RBAR) bulk data card
- Use of PCOMP card to model offset plate as unsymmetric laminate with very soft layer (value of E 2 to 3 orders of magnitude less than plate) serving as the offset space[5]

However, the use of the Z0 field is sufficient for most users to model plate offsets. The result of offsetting a plate depends on the loading condition. For out-of-plane loading (as in the examples), the offset has no effect on out-of-plane displacements, but in-plane displacements increase due to the rotational arc of the element. For in-plane loading, displacements are affected both in-plane and out-of-plane due to the combination of in-plane loading plus offset acting as a moment as well as rotational effects. Note that membrane/bending coupling will play an important part in the correct formulation of the problem, so material cards referenced by offset plates must be provided for both membrane and bending stiffness.

The user must be aware of the differences in the definition of the offset between the CQUAD4, PSHELL and PCOMP cards. The offset value that is used in the Z0 field on the CQUAD4 and PSHELL cards is the distance from the grid point surface to the element mid-plane of the CQUAD4 element. However, on the PCOMP card, the distance appearing on the Z0 field is measured from the grid point surface to the bottom surface of the CQUAD4 element. Also, the Z0 value may be positive or negative depending on the node numbering scheme (clockwise = negative Z0, counterclockwise = positive Z0) and the position of the CQUAD4 element relative to the grid point plane (element above grid point plane = positive Z0, element below grid point = negative Z0). Please note that this is different from what is documented in the User's Manual as of 3/3/90, which properly states offset definition for the PCOMP card only. See Figures 1 and 2 for further detail.

DIFFERENCES BETWEEN COSMIC/NASTRAN, CSAR/NASTRAN AND MSC/NASTRAN

As mentioned in the previous discussion of QUAD4 theory, the theoretical formulation of QUAD4 elements is different in different versions of NASTRAN. COSMIC/NASTRAN and ASTROS share the same QUAD4 element so results compare favorably between these two codes. The COSMIC/NASTRAN QUAD4 element tends to be slightly stiffer and exhibits a closer relationship to linear theory than CSAR and MSC QUAD4 elements. However, all codes give results that compare within 3% of empirical solutions.

EXAMPLE PROBLEMS

1. CANTILEVER PLATE

The cantilever plate problem consists of a semi-monocoque-like structure of plates (QUAD4 elements) attached to a bar (CBAR element) (see Figure 3). The structure is fixed at the wall and has a plane of symmetry on the left side. The cantilever plate can be modeled with the grid points running down the center of the CQUAD4 elements and the bar offset, with the grid points running down the center of the CBAR elements and the plates offset, or with the grid point plane separate and both the CQUAD4 and CBAR elements offset. The result of each of these three methods should compare to each other favorably. These results are located in Table 1.

Table 1

Maximum Displacements
z-displacements
x-displacements

CASE	COSMIC	CSAR	MSC
A. Cantilever Plate Offset on CQUAD	-7.741E-2 -1.963E-3	-7.69E-2 -1.961E-3	-7.76E-2 -1.961E-3
B. Cantilever Plate Offset on CBAR	-7.741E-2 +4.007E-4	-7.701E-2 +4.007E-4	-7.771E-2 +4.006E-4
C. Cantilever Plate CQUAD, CBAR Offset	-7.741E-4 -3.336E-2	-7.669E-2 -3.332E-2	-7.740E-2 -3.327E-2
D. Cantilever Plate Offset on PSHELL	-7.741E-2 -1.963E-3	N/A	N/A

Note: CSAR/NASTRAN and MSC/NASTRAN do no offer field on PSHELL card.

2. MODIFIED CANTILEVER PLATE

The cantilever plate problem was modified to examine some accuracy and user features of the offset capability. The first modification of the cantilever plate was to remove the offset entirely. This results in a cross-shaped cross section instead of a t-shaped cross section and as such is expected to give entirely different results (see Figure 4). The second modification to the cantilever plate is a modified load from a distributed load to a point load at the end of the bar. This gives us a configuration that can be easily compared to an empirical solution (see Figure 5). The third modification to the cantilever plate problem is to change the height

of the bar so that a "stepped" cantilever plate results (see Figure 6). This is to display the interaction of the Z0 fields on the CQUAD4 and PSHELL cards. The results are located in Table 2.

Table 2

Maximum Displacements
z-displacements
x-displacements

CASE	COSMIC	CSAR	MSC
A. Cantilever Plate No offset	-2.794E-1 0.0	-2.789E-1 0.0	-2.794E-1 0.0
B. Cantilever Plate Theory=3.334E-2	-3.400E-2 (1.8% error)	-3.399E-2 (1.8% error)	-3.413E-2 (2.1% error)
C. Cantilever Plate Stepped config.	4.636E-2 -1.385E-3	N/A	N/A

The results from case A show that the cantilever plate run in example 1 with offsets removed show that the configuration is changed and the results are significantly different. This verifies that the offsets used in example 1 are indeed working and giving excellent results. The results from case B show that the cantilever plate with CQUAD4 offset and a point load on the tip of the structure give very close correlation with a theoretical solution of a T-shaped bar of the same dimensions. The results in case C show that placing a standard offset in the Z0 field on the PSHELL card is an efficient method to model a structure where many plates are offset by the same distance. The Z0 field on the PSHELL card can be overridden by an entry on the CQUAD4 card when a few have different offsets (the alternative method is to place an entry in EVERY CQUAD4 card, which can be quite laborious and unnecessary for a large model).

3. CLAMPED PLATE
Note: This problem derived from "Theory of Plates & Shells", by Timoshenko and Woinowsky-Krieger, P.206 (Reference 7)

The clamped plate model is a plate that is clamped on all four sides. Due to the symmetric nature of the structure, only 1/4 of the structure is modeled. There are no elements except CQUAD elements in this model. Three model densities are examined, a 3x3 grid, a 6x6 grid, and a 12x12 grid (see Figure 7). The model is tested with no offset and with a 1.0" offset. According to Reference 7, the empirical solution for this model is -8.806E-4 (no offsets are considered). The results are located in Table 3.

Table 3

Maximum Displacements
z-displacements

CASE	COSMIC	CSAR	MSC
A. Clamped Plate 3x3 grid, no offset	-8.499E-4	-8.95E-4	-8.776-4
B. Clamped Plate 6x6 grid, no offset	-8.743E-4	-9.00E-4	-8.923E-4
C. Clamped Plate 12x12 grid, no offset	-8.802E-4	-8.962E-4	-8.874E-4
D. Clamped Plate 3x3 grid, 1.0 offset	-8.499E-4	-1.478E-4	-8.961E-5
E. Clamped Plate 6x6 grid, 1.0 offset	-8.743E-4	-2.885E-4	-1.154E-4
F. Clamped Plate 12x12 grid, 1.0 offset	-8.802E-4	-5.515E-4	-2.639E-4

The results show that, in the no offset case, the COSMIC QUAD4 element is slightly stiffer and exhibits better correlation with linear theory as it asymptotically approaches the empirical solution. All cases, however, compare well with the empirical solution. In the offset cases, the reason for great differences in CSAR/NASTRAN and MSC/NASTRAN cannot be explained.

4. SANDWICH PLATE

The sandwich plate models 1/4 of a symmetric plate structure with all four edges constrained in the out-of-plane direction and a loading in the center of the symmetric section of the plate (see Figure 8). It is modeled using metal and composite sandwich plates. The metal sandwich plates are modeled using a separate material card to specify transverse shear properties. The composite sandwich plates are modeled in a two step process, first using a PCOMP card to input the properties of the composites, then from the output the equivalent properties as PSHELL/MAT2 cards is extracted, and rerun with modified PSHELL and MAT2 cards. This procedure is described at length in reference 4. Results are located in Table 4.

Table 4

Maximum Displacments
z-displacements
x-displacements

CASE	COSMIC	CSAR	MSC
A. Metal Sandwich No Offset	-3.747E-2 0.0	-3.770E-2 0.0	-3.792E-2 0.0
B. Metal Sandwich Offset on CQUAD	-3.72E-2 -5.00E-3	-2.960E-2 -2.406E-6	-3.690E-2 -1.520E-2
C. Composite Sandwich No Offset	-2.667E-2 0.0	-2.696E-2 0.0	-2.721E-2 0.0
D. Composite Sandwich Offset on CQUAD	-2.626E-2 +1.365E-4	-1.394E-2 +4.131E-7	-2.663E-2 +1.391E-4

The results show that both metal sandwich and composite sandwich plates can be modeled with and without offset. The reason for different results from CSAR/NASTRAN for offset cases cannot be explained.

5. LAMINATED PLATE

The laminated plate model is identical to the cantilever plate model (see Figure 9). The difference is that in this case both metal and composite laminated plates are used in place of the isotropic plate used in example 1. The problem is run with the CBAR offset from the CQUAD4 and with the CQUAD4 offset from the CBAR with the CQUAD4 offset on the PCOMP card. The reason that the Z0 field was used on the PCOMP card rather than the CQUAD4 card is that the Z0 field on the CQUAD card, when used in conjunction with a PCOMP card, appears to be inactive for both COSMIC/NASTRAN and CSAR/NASTRAN. It is operating in MSC/NASTRAN. Note that this is not the case for the CQUAD4/PSHELL combination, where the Z0 can be used on either card. Results are located in Table 5.

Table 5

Maximum Displacements
z-displacements
x-displacements

CASE	COSMIC	CSAR	MSC
A. Metal Laminate PCOMP Offset	-3.406E-2 -9.786E-4	-3.319E-2 -9.686E-4	-3.404E-2 -9.741E-4
B. Metal Laminate CBAR Offset	-3.406-2 +1.979E-4	-3.386E-2 +1.978E-4	-3.410E-2 +1.978E-4
C. Composite Laminate PCOMP Offset	-6.250E-2 -1.030E-3	-6.250E-2 -1.030E-3	-6.060E-2 -1.030E-3
D. Composite Laminate CBAR Offset	-6.250E-2 +1.150E-3	-6.250E-2 +1.150E-3	-6.230E-2 +1.150E-3

The results show that laminated plates, both metal and composite, can be accurately and easily modeled using offset capabilities.

CONCLUSION

The results of studies performed in this paper indicate that the offset feature provided in COSMIC/NASTRAN for the QUAD4/TRIA3 elements is performing as expected. The results are compared against empirical solutions and other NASTRAN variations (MSC and CSAR). These results generally show excellent agreement except in some comparisons with MSC and CSAR, where COSMIC results appear to be correct.

REFERENCES

1. COSMIC/NASTRAN User's Manual, Volume 1
2. MSC/NASTRAN User's Manual, Volume 1
3. CSAR/NASTRAN User's Manual, Volume 1
4. Venkayya, V.B., Tischler, V.A., QUAD4 Seminar, WRDC-TR-89-3046, April 1989
5. CSAR/NASTRAN User Newsletter, 4th Quarter 1991
6. MSC/NASTRAN Application Manual, Volume 2
7. Timoshenko & Woinowsky-Krieger, Theory of Plates and Shells, McGraw-Hill, 1952.

QUAD4 - OFFSET DEFINITION

(CQUAD4 PSHELL)

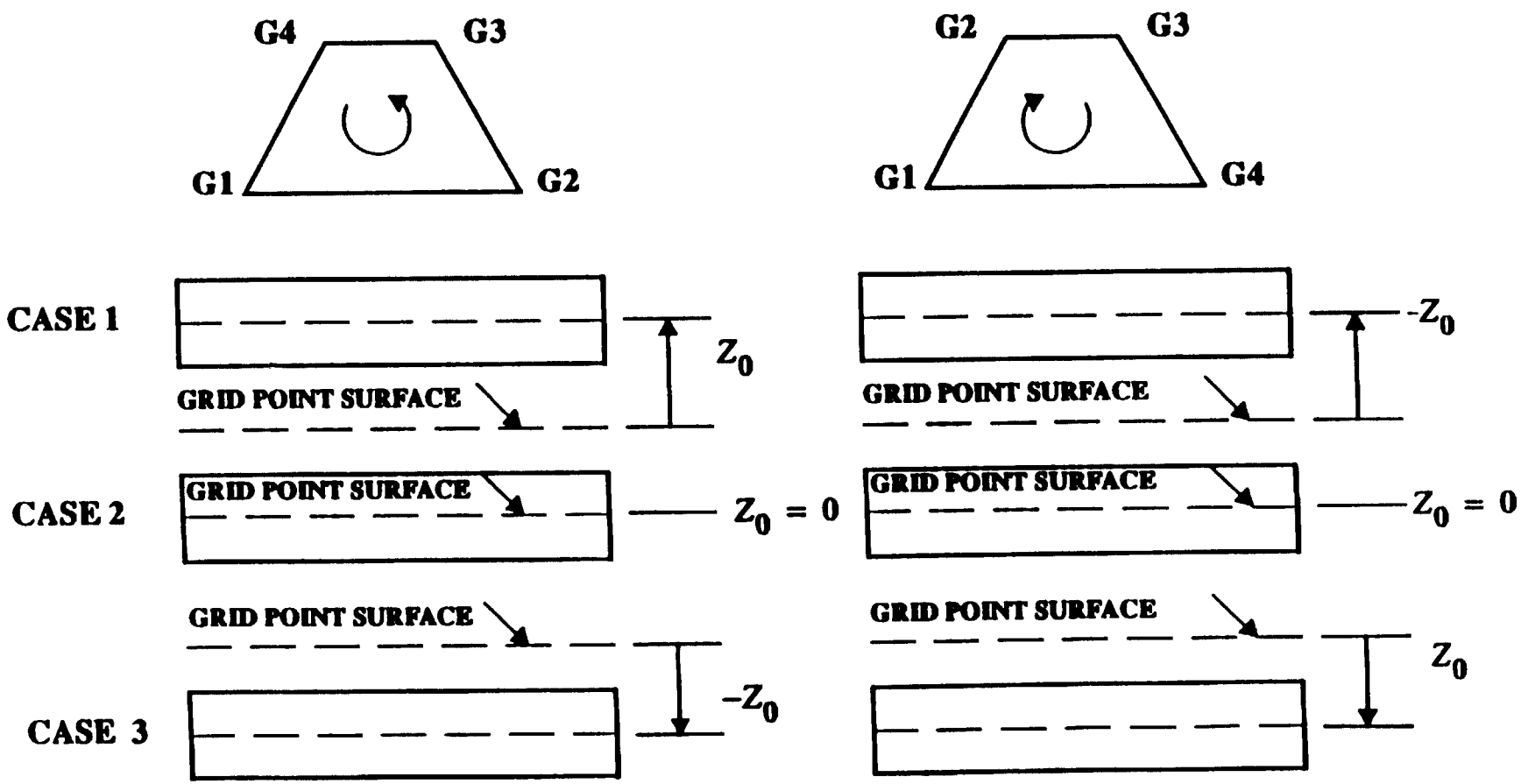


Figure 1

QUAD4 - OFFSET DEFINITION

(PCOMP PCOMP1 PCOMP2)

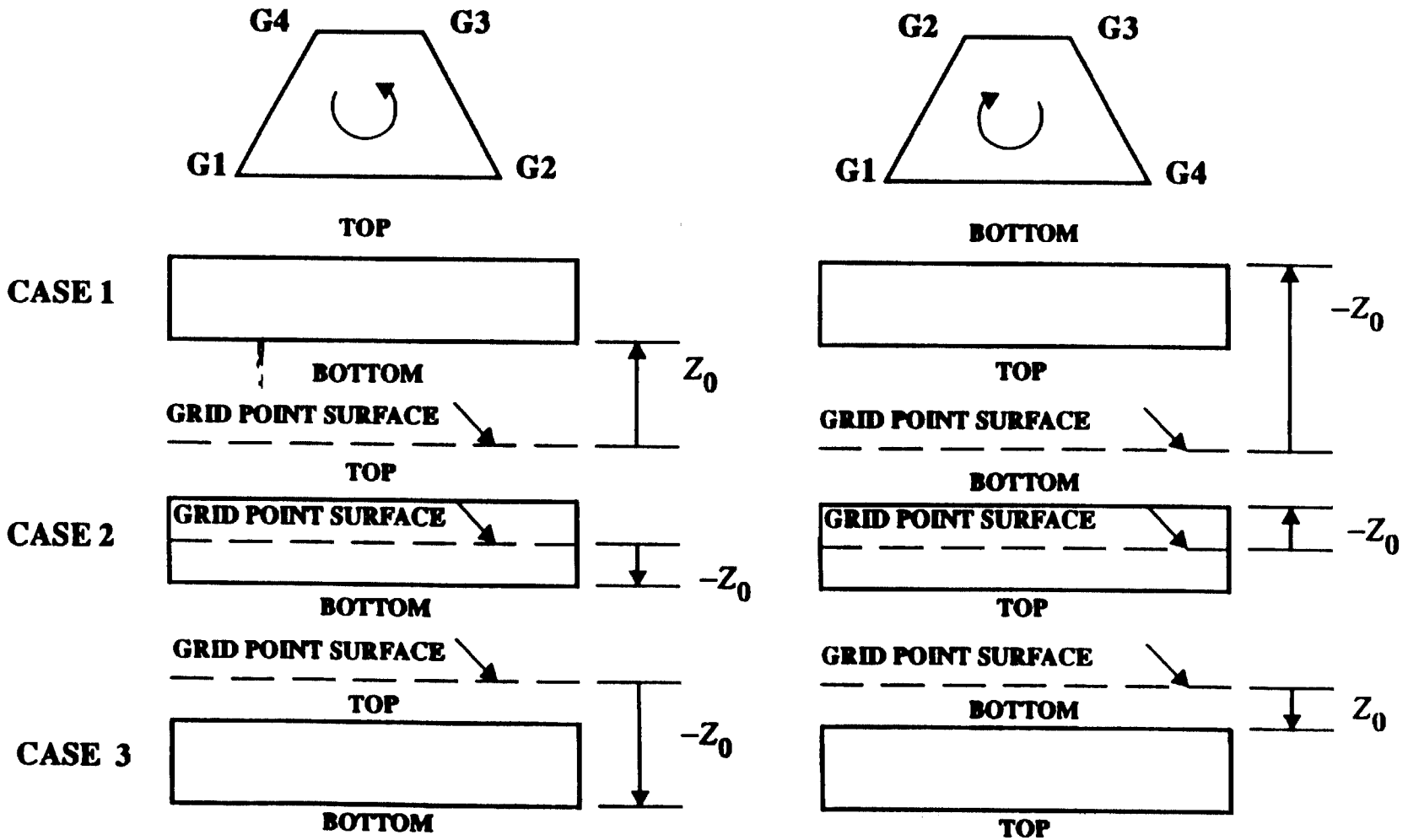
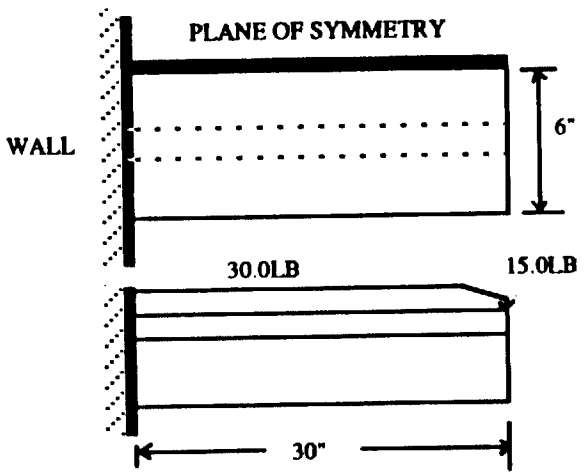


Figure 2

RESULTS

CANTILEVERED PLATE



MATERIAL PROPERTIES

E = 29.0E+6

NU = .3

I 1 (BAR) = .042

I 2 (BAR) = .0104

J(BAR) = .028625

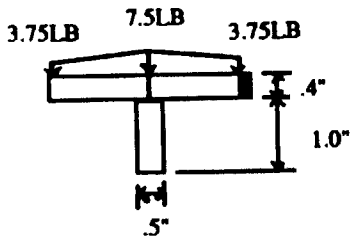
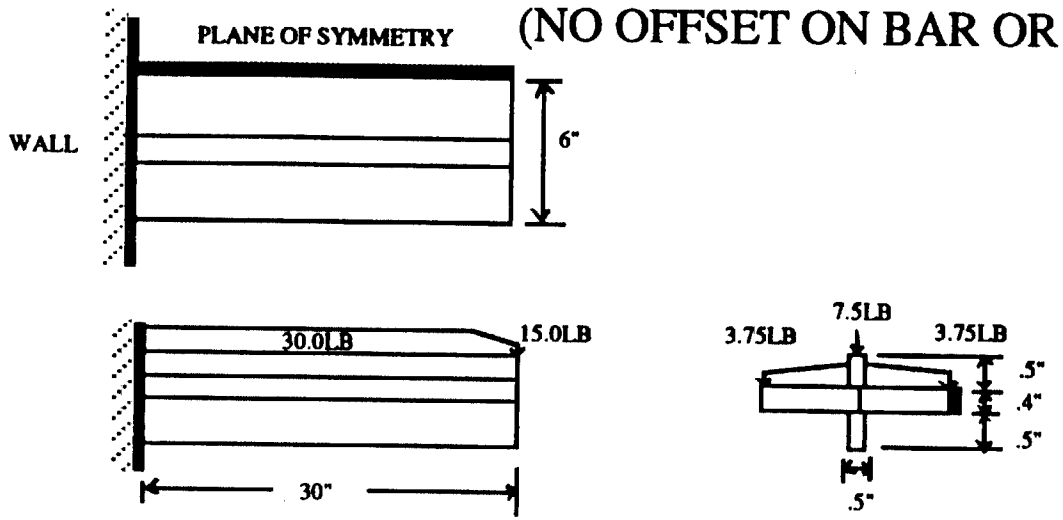


Figure 3

RESULTS

MODIFIED CANTILEVERED PLATE
(NO OFFSET ON BAR OR PLATE)



MATERIAL PROPERTIES

E = 29.0E+6

NU = .3

I 1 (BAR) = .042

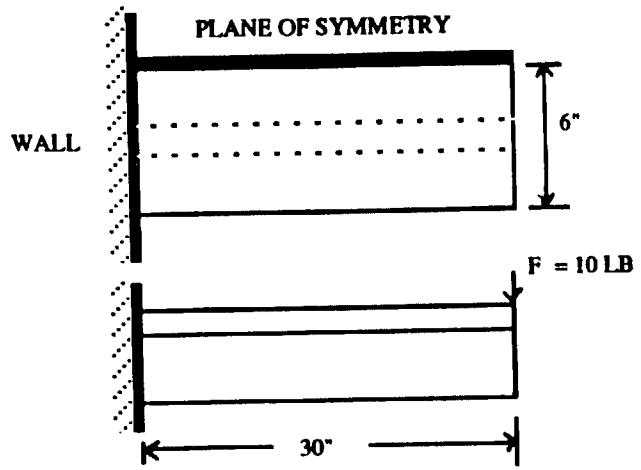
I 2 (BAR) = .0104

J(BAR) = .028625

Figure 4

RESULTS

MODIFIED CANTILEVERED PLATE
(NEW LOADING CONDITION)



MATERIAL PROPERTIES

E = 29.0E+6

NU = .3

I 1 (BAR) = .042

I 2 (BAR) = .0104

J(BAR) = .028625

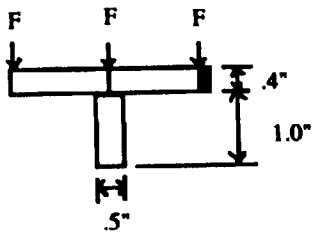


Figure 5

RESULTS

MODIFIED CANTILEVERED PLATE
(STEPPED PLATE)

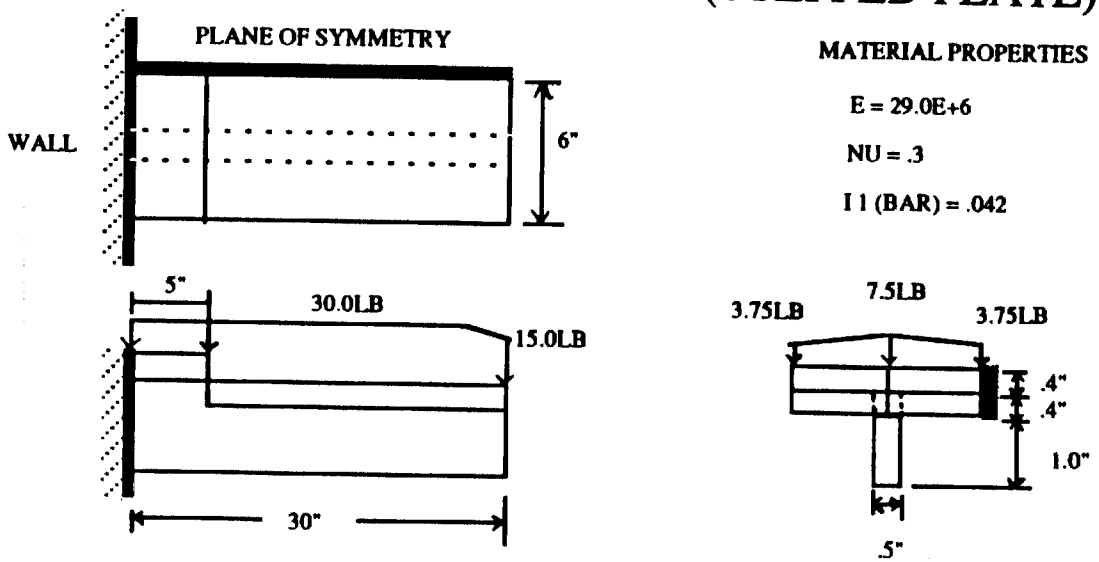
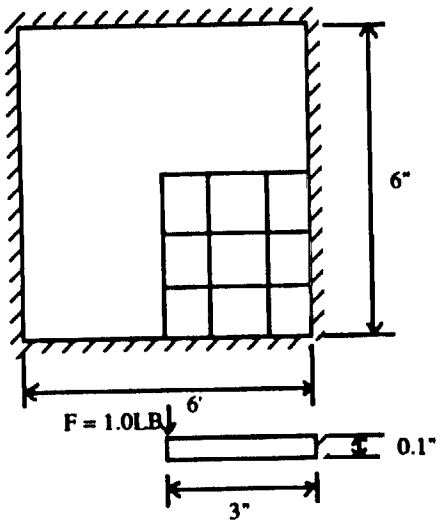


Figure 6

RESULTS

SIMPLE PLATE (CLAMPED)



MATERIAL PROPERTIES

E = 10.0 E +6

NU = .3

12I / T3 = 1.0

GRID MODELED AS 3X3, 6X6, AND 12X12

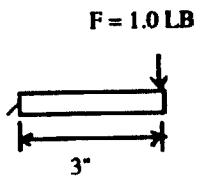
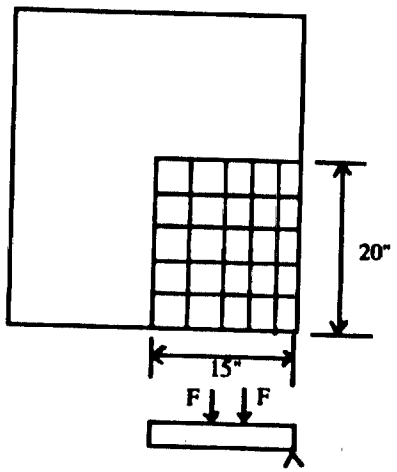


Figure 7

RESULTS

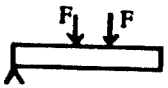
SANDWICH PLATE



SIMPLY SUPPORTED BOUNDARY CONDITION

MATERIAL PROPERTIES:

F = 1000 LB



METAL SANDWICH PROPERTIES

MAT1, 22, 10.0E+6, , .3

MAT1, 23, , 3.0E+4

PSHELL, 72, 22, .3776, 22, 31.2355, 23, 2.6483

COMPOSITE SANDWICH PROPERTIES

MAT8, 22, 18.5E+6, 1.6E+6, .25, .65E+6, 4.0E+5, .005

PCOMP, 72, .5, , , , , +ABC

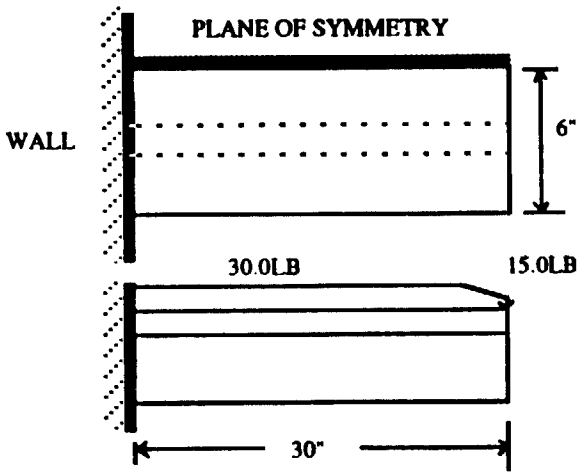
+ABC, 22, .0236, 45., YES, 22, .0236, -45., YES, +BCD

+BCD, 22, .0236, 90., YES, 22, .0236, 0., YES

Figure 8

RESULTS

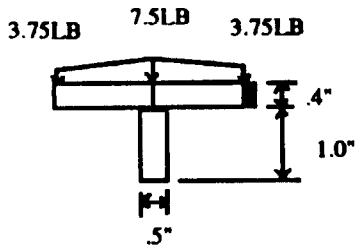
LAMINATED PLATE



MATERIAL PROPERTIES:

METAL LAMINATE

E = 29.0E+6
NU = .3
RHO = 7.324E-4
SBOND = .65E+6
T = 4.1" PLIES



COMPOSITE LAMINATE

PCOMP SAME AS METAL LAMINATE

MAT8, 70, 1.85E+7, 1.6E+6, .25, .65E+6,,,.055,+MT5

+MT5, 0., 0., 100., 1.15E+5, 1.15E+5, 1.15E+5, 1.15E+5, 1.0E+15

NOTE: Z0 FIELD ON CQUAD4 WHEN USED WITH PCOMP IS INOPERABLE (CQUAD4/PSHELL OK)

Figure 9

Eigenvalue Routines in NASTRAN*

A Comparison with the Block Lanczos Method

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:

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

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.

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

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.

$$\lambda_i = \frac{x_i^T K x_i}{x_i^T M x_i} = \frac{K_{ii}}{M_{ii}} \tag{5}$$

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

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

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

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

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.

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

where λ_i, $i = 1, 2 \ldots n$ 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:

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

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

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

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

Recall that an orthogonal transformation is one whose matrix T satisfies

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

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

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

then from Eq (22)

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

i.e.

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

by repeatedly applying Eq (23). Eq (25) implies that y is an eigenvector of the transformed matrix $T_r T_{r-1} \cdots T_2 T_1 A T_1^T T_2^T \cdots T_{r-1}^T T_r^T$. 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

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

Since Q_r is orthogonal, then

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

In the limit as $r \to \infty$ 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

$$Bx = \Lambda M x \tag{31}$$

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.

$$Bx = \Lambda Mx$$

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

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

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

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

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

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.

$$V = [V_1, V_2, \ldots V_m] \qquad (39)$$

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.

$$V^T M V = I \qquad (40)$$

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

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

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

$$x = V y \tag{45}$$

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 σ = .046037. Note that the smallest 8 eigenvalues are transformed from closely spaced eigenvalues to eigenvalues with good separation.

Satellite Problem

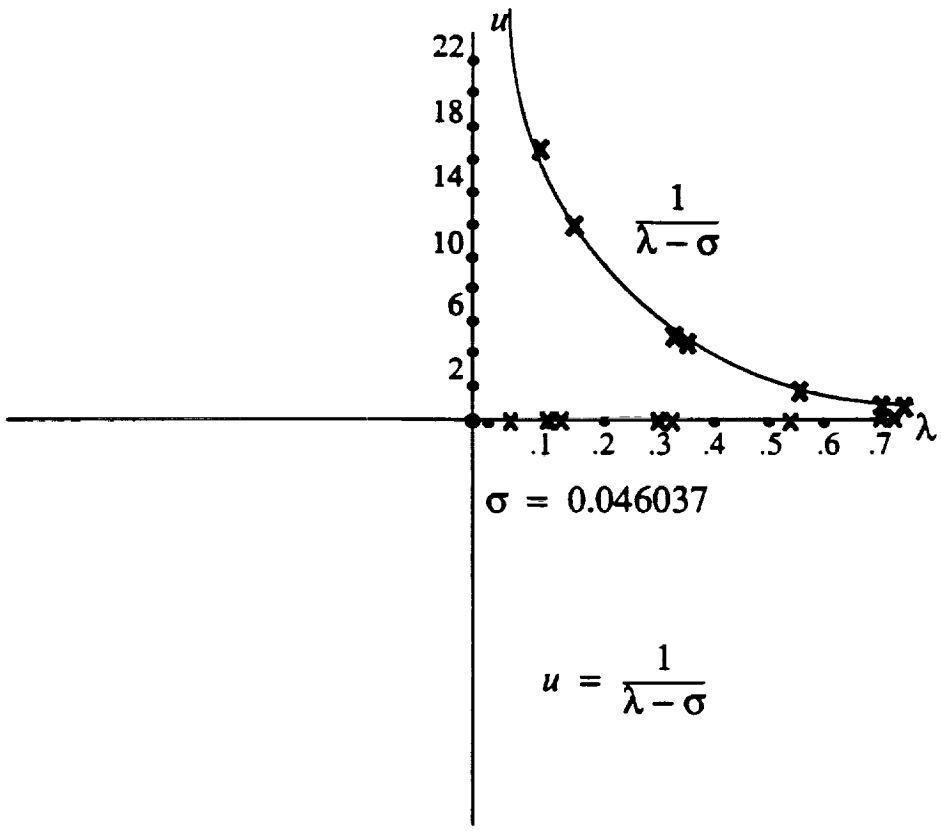


FIGURE 1

	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, $Q_1, Q_2, \ldots Q_j$, 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

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

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

$$W_j^T H W_j = W_j^T W_j T_j + W_j^T Q_{j+1} B_{j+1} E_j^T$$

i.e.

$$W_j^T H W_j = T_j \tag{54}$$

since

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

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. $T_j s = s\theta$, then $(\lambda, W_j s)$ 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.

$$M(K - \sigma M)^{-1} M x = u M x$$

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

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

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

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

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

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

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

Therefore $|\lambda-\nu| \leq \frac{\beta_j}{\theta^2}$. 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 algo-

rithms 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.

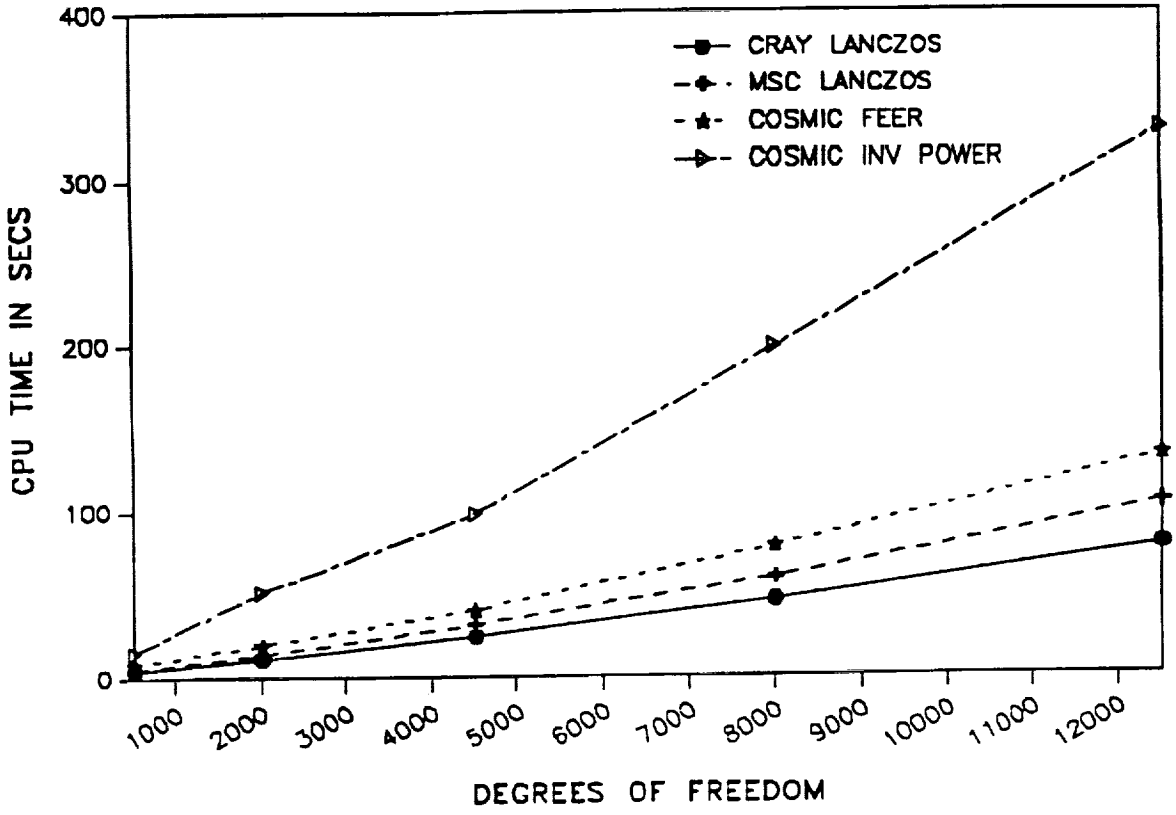


Figure 2: Degrees of Freedom versus CPU Time in Seconds.

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 $p = 7$.

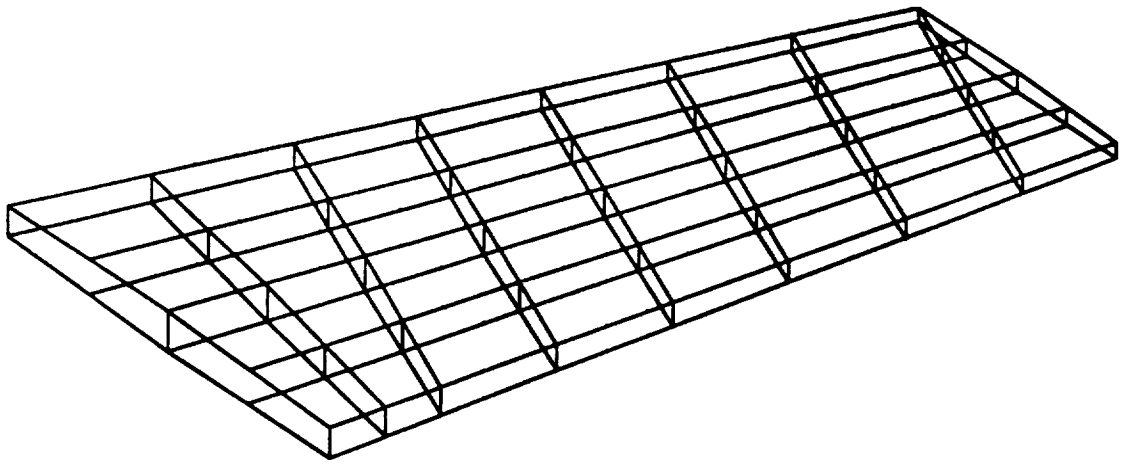


Figure 3: Intermediate Complexity Wing

	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 $p = 7$.

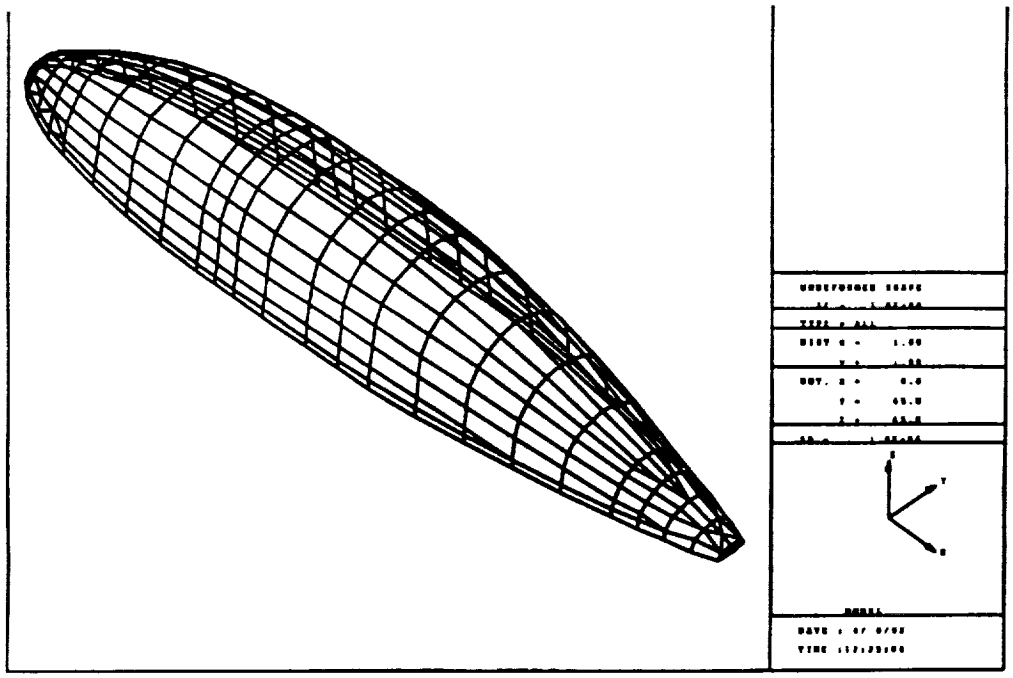


Figure 4: Radome

	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.

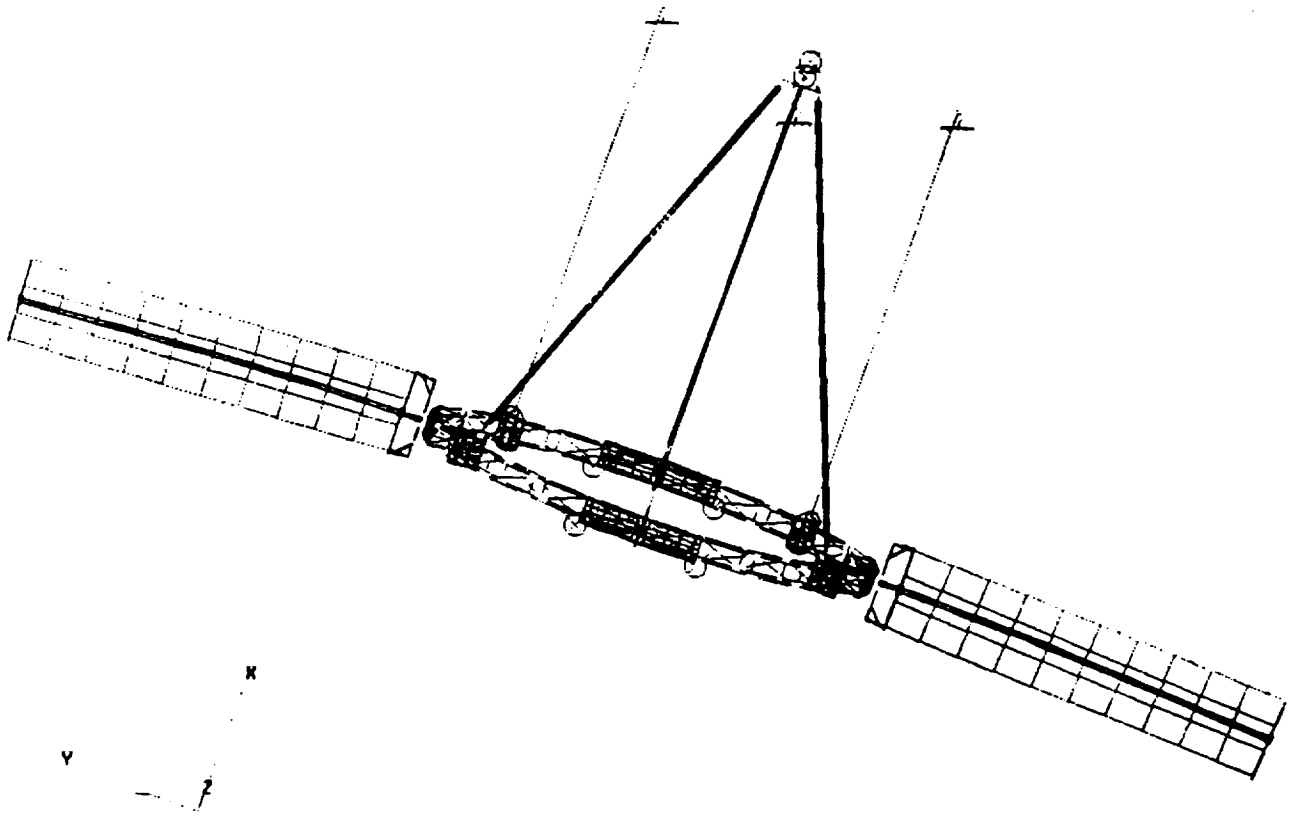


Figure 5: Satellite

	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 $p = 7$.

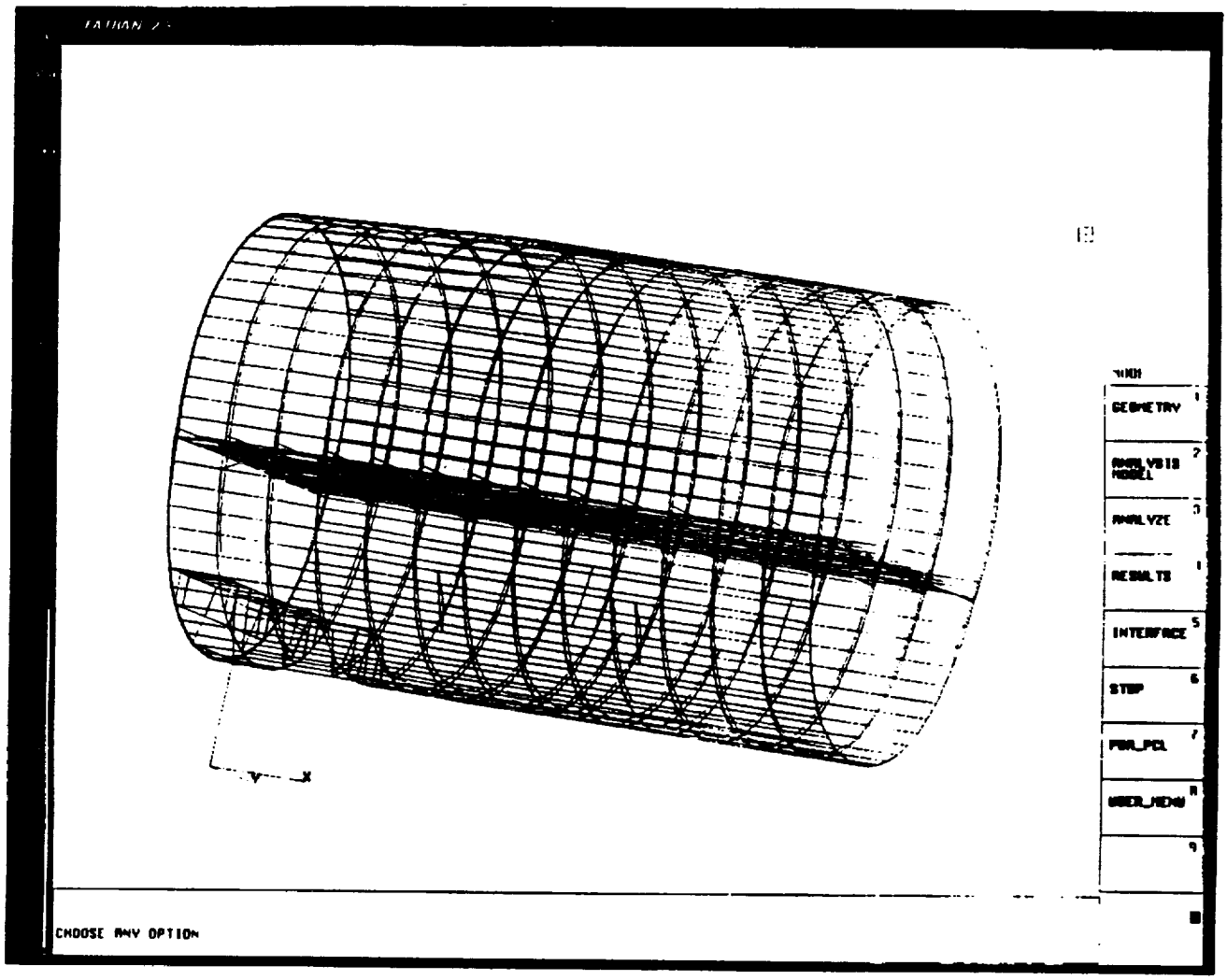


Figure 6: Forward Fuselage

	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													CPU TIME IN SECS
	1	5	10	15	20	25	30	35	40	45	50	55	59	
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.

AUTOMATIC ASET SELECTION FOR DYNAMICS ANALYSIS

Tom Allen

McDonnell Douglas Space Systems Company - Huntsville Division

ABSTRACT

A method for selecting optimum NASTRAN analysis set degrees of freedom for the dynamic eigenvalue problem is described. Theoretical development of the Guyan reduction procedure on which the method is based is first summarized. The algorithm used to select the analysis set degrees of freedom is then developed. Two example problems are provided to demonstrate the accuracy of the algorithm.

1.0 INTRODUCTION

A NASTRAN user is faced with two major difficulties when solving a dynamic eigenvalue problem. First, an eigenvalue solution is expensive for most structural problems encountered in engineering applications. Second, many more degrees of freedom (DOF) are required to define a structure's elastic properties than are required to define its inertial properties, which tends to exacerbate the first difficulty.

A popular method for easing the severity of these difficulties is to reduce the problem size using Guyan reduction (Reference 1). This method allows the user to preserve the elastic properties of the reduced problem set while reducing the problem size to one more manageable for a dynamic eigenvalue solution. At the same time, the mass properties are also condensed with some penalty associated with the reduction of mass from the coordinates being eliminated. The present paper describes an approach for optimizing the partitioning process to minimize this penalty.

Theoretical development of the Guyan reduction method is presented in Section 2. Section 3 describes the algorithm used to select automatically the analysis set degrees of freedom. Verification of the method is presented in Section 4. Conclusions are presented in Section 5.

2.0 THE GUYAN REDUCTION METHOD

By way of introduction, the Guyan reduction method will first be reviewed.

The dynamic eigenvalue problem is given by the equation

$$([K] - \lambda[M])\{\phi\} = 0 \qquad (1)$$

where K is the structural stiffness matrix, M is the structural mass matrix, λ is the eigenvalue, and ϕ is the eigenvector or modal displacements. The Guyan reduction method starts by partitioning Equation 1 into independent DOF, designated in NASTRAN as the A-set, and dependent DOF, designated as the O-(for OMIT) set. After performing this operation Equation 1 becomes

$$\left(\begin{bmatrix} \bar{K}_{aa} & K_{ao} \\ K_{ao}^T & K_{oo} \end{bmatrix} - \lambda \begin{bmatrix} \bar{M}_{aa} & M_{ao} \\ M_{ao}^T & M_{oo} \end{bmatrix} \right) \begin{Bmatrix} \phi_a \\ \phi_o \end{Bmatrix} = 0 \tag{2}$$

where the subscript "a" denotes A-set DOF and the subscript "o" denotes O-set DOF.

A set of constraints for the O-set displacements can be derived by solving for ϕ_o in terms of ϕ_a using statics, or

$$K_{ao}^T \phi_a + K_{oo} \phi_o = 0 \tag{3}$$

The O-set displacements now become

$$\phi_o = G_o \phi_a \tag{4}$$

where

$$G_o = -K_{oo}^{-1} K_{ao}^T \tag{5}$$

Equation 4 defines ϕ_o as the deflections at O-set DOF due to unit displacements at the A-set DOF. Stated another way, the O-set displacements, ϕ_o, are constrained to move in relation to A-set displacements, ϕ_a, as governed by the transformation matrix G_o. This relationship constitutes a Ritz transformation of the eigenvalue problem. The transformation written in terms of the full displacement set is

$$\{\phi\} = \begin{Bmatrix} \phi_a \\ \phi_o \end{Bmatrix} = [G]\{\phi_a\} = \begin{bmatrix} I \\ G_o \end{bmatrix} \{\phi_a\} \tag{6}$$

Using this Ritz transformation the reduced mass and stiffness matrices become

$$[M_{aa}] = [G]^T [M][G] \tag{7}$$

and

$$[K_{aa}] = [G]^T [K][G] \tag{8}$$

Performing these operations on the matrices in Equation 2 we get

$$[M_{aa}] = [\bar{M}_{aa}] + 2[M_{ao}][G_o] + [G_o]^T [M_{oo}][G_o] \tag{9}$$

and

$$[K_{aa}] = [\bar{K}_{aa}] + [K_{ao}][G_o] \tag{10}$$

The mass of the system will be redistributed based on the elastic connections between the O-set DOF and the A-set DOF as shown in Equation 9.

Note that Guyan reduction is exact when M_{oo} (and hence M_{ao}) is a null matrix and gives the best solution for any selected partition when it is not. It does not, however, address directly the problem of selecting most effectively the set of independent DOF that will best serve the aims of the user. For this, a means of removing terms from the mass matrix so as to minimize the impact on the solution accuracy must be determined.

3.0 ASET SELECTION ALGORITHM

As stated previously, Guyan reduction is exact when M_{oo} is null, or when the O-set mass to stiffness "ratio" is zero. As the mass to stiffness "ratio" between M_{oo} and K_{oo} increases, the accuracy of the Guyan reduction method decreases. This generalization forms the basis of the A-set selection method.

The six step method for determining the A-set DOF is as follows:

1. Execute NASTRAN to obtain an initial M_{aa}, K_{aa}, and A-set table. The mass and stiffness matrices can be reduced as desired in NASTRAN as long as the modal content over the frequency range of interest is retained. Note that no reduction need be performed at this stage but the initial constraint equation must be applied.
2. Define the number of DOF that will be in the final A-set. These DOF may also contain a "kernel" set of DOF that will remain in the A-set regardless of their mass to stiffness ratio.
3. Determine the minimum mass to stiffness "ratio" for the O-set DOF. Because M and K are diagonally dominant, this ratio is most easily approximated by stripping the diagonal from M and K and scanning for the minimum M_{ii}/K_{ii} which we will call min(M/K). The min(M/K) DOF is then partitioned from M and K and reduced from the system, provided it is not a member of the kernel set.
4. Repeat step 3 until the desired number of DOF remain in the A-set.
5. Write NASTRAN ASET bulk data cards for the retained DOF
6. Check the A-set to determine if desired modes are adequately defined.

To improve the efficiency of the check process, the mass and stiffness matrices may be saved during Step 5. These matrices can then be used in an eigenvalue analysis to determine if the selected A-set is adequate.

The user may, if desired, decide to refine the A-set further if it is concluded that more DOF can be reduced from the problem. To simplify this second reduction, the A-set listing and matrices from Step 5 can be used as input to Step 2. The process would then proceed as before.

Occasionally, too few DOF will be defined in the A-set. By keeping track of the DOF placed in the O-set during each iteration, the user may simply review DOF that were omitted during previous iterations to determine DOF that are required to define the mode or modes lost because of the Guyan reduction. He may then selectively include those DOF deemed necessary to the A-set by adding these DOF on his ASET bulk data cards. Alternatively, he may save intermediate ASET card images for convenience.

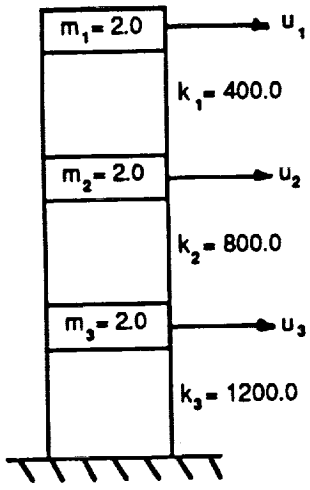
Because the algorithm currently works on one DOF at a time, the user should use NASTRAN to make the problem size as small as possible to decrease the solution time. Though reducing several DOF during each iteration is a desirable feature, no definitive method for including this feature in the algorithm has yet been developed. More information on this topic is presented in the conclusions.

The algorithm described above virtually guarantees that the smallest A-set will be obtained with minimal effort, provided that too severe a reduction is not specified. The general procedure for selecting the A-set automatically should be clear from the discussion above. The process is best illustrated, however, by performing sample calculations on a simplified model, as shown in the next section.

4.0 METHOD VERIFICATION

Two sample problems were developed to validate the A-set selection method. The first problem is a simplified model of a three story building. The reduction operations are performed by hand to clarify the algorithm. The second sample problem determines the A-set of a 3600 DOF NASTRAN model. The A-set for this problem was generated using a program developed by McDonnell Douglas Space Systems Company-Huntsville Division (MDSSC-HSV). The data from these sample problems verify the algorithm outlined in Section 3.

The simplified model of the three story structure is shown in Figure 1. The mass and stiffness matrices are also shown. The fundamental frequency of this system is 1.45 Hz. We want to reduce the problem to a one DOF system.



$$[M] = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \qquad \{\phi_1\} = \begin{Bmatrix} 1.0 \\ 0.585 \\ 0.255 \end{Bmatrix}$$

$$[K] = \begin{bmatrix} 400 & -400 & 0 \\ -400 & 1200 & -800 \\ 0 & -800 & 2000 \end{bmatrix}$$

$$f_1 = 1.45 \text{ Hz}$$

Figure 1. Simplified Three Story Building

First we find the min(M/K) for this system which is 2/2000 = 0.001 for displacement u_3. Partitioning this DOF from M and K yields

$$\bar{M}_{aa} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \qquad M_{ao} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad M_{oo} = 2$$

$$\bar{K}_{aa} = \begin{bmatrix} 400 & -400 \\ -400 & 1200 \end{bmatrix} \qquad K_{ao} = \begin{bmatrix} 0 \\ -800 \end{bmatrix} \qquad K_{oo} = 2000$$

The G_o matrix for this problem is

$$G_o = \frac{-1}{2000} [\, 0 \quad -800 \,] = [\, 0.0 \quad 0.4 \,]$$

The reduced mass and stiffness matrices are found using Equations 9 and 10 and are

$$M_{aa} = \begin{bmatrix} 2 & 0 \\ 0 & 2.32 \end{bmatrix}$$

$$K_{aa} = \begin{bmatrix} 400 & -400 \\ -400 & 880 \end{bmatrix}$$

We repeat the steps to determine the mass and stiffness of the one DOF system. Performing these steps produces M = 2.48 and K = 218.2. The frequency for this one DOF system is f_1 = 1.50 Hz which is 3.5 percent higher than the "exact" frequency of 1.45 Hz.

Though the frequencies show excellent agreement, correlation between the mode shapes should also be verified. Back transforming using G_o we get

$$\{\phi_1\} = \begin{Bmatrix} 1.0 \\ 0.455 \\ 0.182 \end{Bmatrix}$$

for the one DOF system. We will use the modal assurance criterion (MAC) described in Reference 2 to measure the correlation between this mode shape and the "exact" mode shape. The MAC between any two modes varies from zero, meaning no correlation, to unity, meaning perfect correlation. The MAC for these modes is 0.987 indicating that little modal accuracy was lost during the reduction.

The second sample problem involves finding an A-set for the model shown in Figure 2. The unreduced model has approximately 3600 DOF. Currently, the model A-set has 180 DOF which was used as a starting point for this problem. This A-set was further reduced to 50 DOF using the MDSSC-HSV developed program based on the selection algorithm described in Section 3. The final A-set size is approximately 25 percent of the original A-set size.

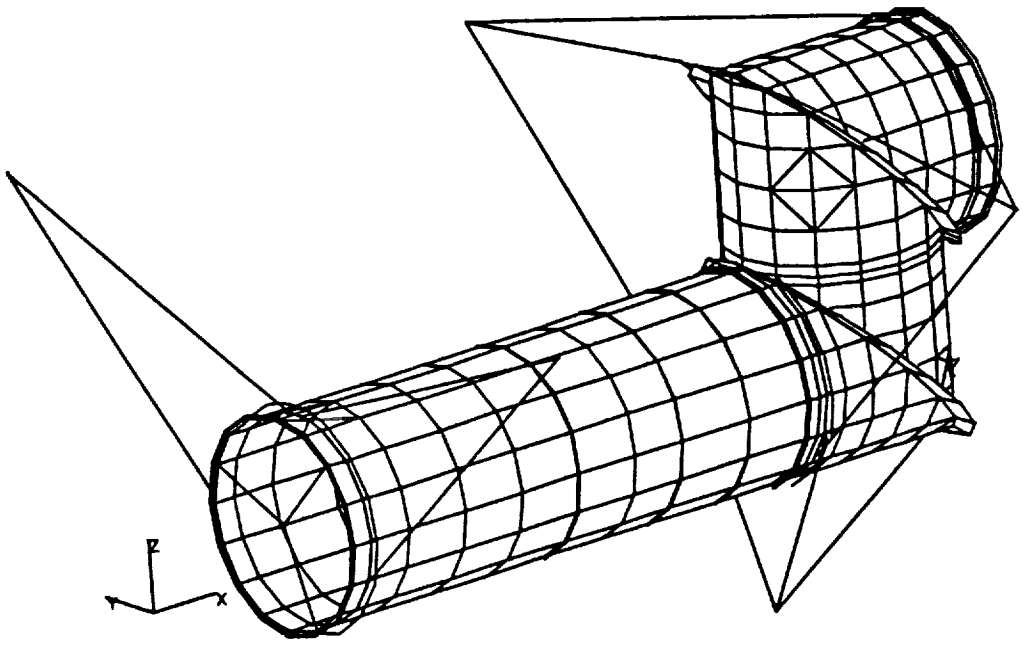


Figure 2. NASTRAN Model for Sample Problem 2

Table 1 shows a comparison between the frequencies and mode shapes of the 180 DOF model and the 50 DOF model. The frequencies show excellent agreement with a maximum difference of 1.4 percent for the sixth mode. The mode shapes are almost perfectly correlated between the the 180 DOF model and the 50 DOF model. Indeed, it may be possible to reduce the problem size even further.

Table 1. Frequency and Mode Shape Comparison Between 180 DOF Model and 50 DOF model

MODE	f_{180}	f_{50}	Δ%	MAC
1	11.9	11.9	0.0	>0.999
2	12.9	13.0	0.8	>0.999
3	24.1	24.2	0.4	0.998
4	24.9	25.0	0.4	0.996
5	33.1	33.3	0.6	0.998
6	62.3	63.2	1.4	0.992

5.0 CONCLUSIONS

A method for automatically selecting the NASTRAN A-set DOF was described. Theoretical development and an outline of the steps involved were provided. Two example problems were provided that demonstrate the use and the accuracy of the method. Some potential enhancements have been identified and will be briefly summarized here.

One potential enhancement noted earlier would be to reduce multiple DOF during each iteration. Because of the redistribution of the mass of the system, simply reducing a certain percentage of the DOF at each iteration is to be discouraged. The reason for this is best demonstrated with an example.

Consider the simply supported beam of Figure 3. Because all of the DOF have identical mass to stiffness ratios, the removal would begin with the first DOF with this min(M/K). If a 20 percent reduction rate were chosen then u_1 and u_2 would be removed in the first iteration, which could ultimately result in a poorly chosen A-set.

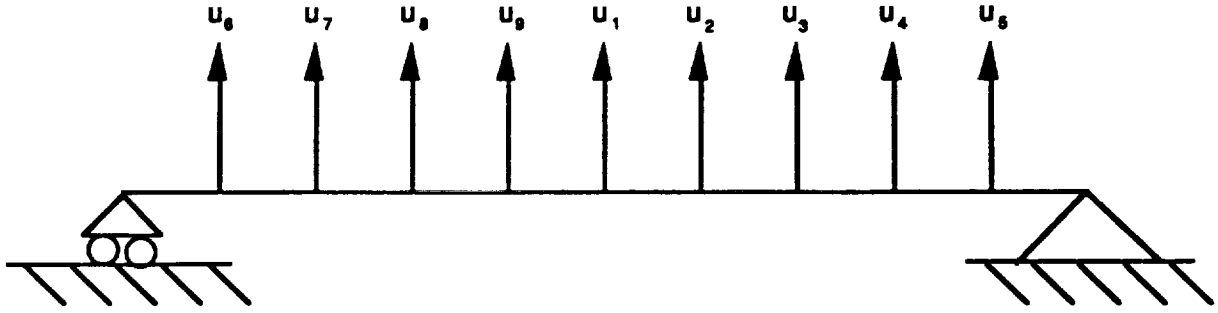


Figure 3. Simply Supported Beam

A second potential enhancement would be including a method in the algorithm that would determine the optimum number of A-set DOF based on a user defined upper bound frequency of interest. Because the algorithm removes terms with a high pseudo frequency, i.e. large K_{ii}/M_{ii}, an approach based on the pseudo frequencies of the reduced system could be used to predict the minimum required number of A-set DOF.

Even without these enhancements, the method has been successfully implemented at MDSSC-HSV. The often tedious, and sometimes error prone A-set selection process has been automated, saving engineering time while increasing A-set efficiency.

REFERENCES

1. Guyan, R.J.: "Reduction of Stiffness and Matrices," AIAA Journal, Volume 3, pg 380, 1965.

2. Ewins, D.J.: Modal Testing: Theory and Practice, John Wiley & Sons Inc., New York, June 1985.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (leave blank)	2. REPORT DATE September 1993	3. REPORT TYPE AND DATES COVERED Conference Publication
4. TITLE AND SUBTITLE Twenty-First NASTRAN® Users' Colloquium		5. FUNDING NUMBERS
6. AUTHOR(S)		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) COSMIC, NASA's Computer Software Management and Information Center The University of Georgia Athens, GA 30602		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CP-3203

11. SUPPLEMENTARY NOTES
Also Available from COSMIC, Athens, GA 30602

12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category - 39	12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

This publication contains the proceedings of the Twenty-First NASTRAN Users' Colloquium held in Tampa, FL, April 26 through April 30, 1993. It provides some comprehensive general papers on the application of finite elements in engineering, comparisons with other approaches, unique applications, pre- and postprocessing with other auxiliary programs and new methods of analysis with NASTRAN.

14. SUBJECT TERMS NASTRAN, structures, structural analysis, finite element analysis, colloquium			15. NUMBER OF PAGES 188 16. PRICE CODE A09
17. SECURITY CLASSIFICATION OF REPORT Unclass	18. SECURITY CLASSIFICATION OF THIS PAGE Unclass	19.SECURITY CLASSIFICATION OF ABSTRACT Unclass	20. LIMITATION OF ABSTRACT Unlimited

Available from NASA Center for AeroSpace Information
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
(301) 621-0390