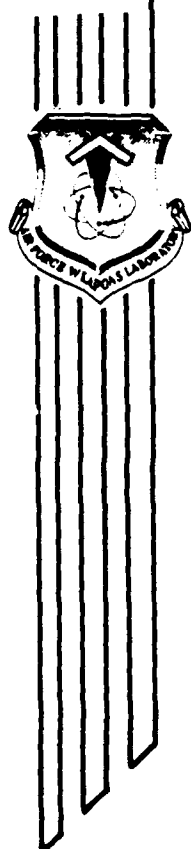


2

AD-A203 447



VAX TO CRAY NASTRAN USER INTERFACE

M. R. James

Applied Technology Associates, Inc.
Albuquerque, NM 87119

DTIC FILE COPY

October 1988

Final Report

Approved for public release; distribution unlimited.

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117-6008

DTIC
ELECTE
JAN 3 1989
S CH D

89 1 03 024

This final report was prepared by Applied Technology Associates, Inc., Albuquerque, New Mexico, under Contract F29601-86-C-0252, Job Order 3005AR07 with the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico. Mr. Ken Qassim (ARBC) was the Laboratory Project Officer-in-Charge.

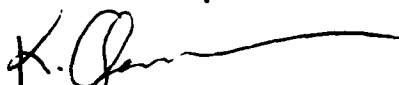
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been authored by a contractor of the United States Government. Accordingly, the United States Government retains a nonexclusive, royalty-free license to publish or reproduce the material contained herein, or allow others to do so, for the United States Government purposes.

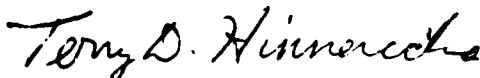
This report has been reviewed by the Public Affairs Office and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nationals.

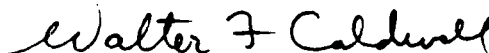
If your address has changed, if you wish to be removed from our mailing list, or if your organization no longer employs the addressee, please notify AFWL/ARBC, Kirtland AFB, NM 87117-6008 to help us maintain a current mailing list.

This report has been reviewed and is approved for publication.


KEN QASSIM
Project Officer

FOR THE COMMANDER


TERRY D. HINNERICHS
Major, USAF
Chief, Control Development Branch


WALTER F. CALDWELL
Lt Colonel, USAF
Chief, Controls Division

DO NOT RETURN COPIES OF THIS REPORT UNLESS CONTRACTUAL OBLIGATIONS OR NOTICE ON A SPECIFIC DOCUMENT REQUIRES THAT IT BE RETURNED.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified			1b RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.		
2b DECLASSIFICATION / DOWNGRADING SCHEDULE					
4 PERFORMING ORGANIZATION REPORT NUMBER(S)			5 MONITORING ORGANIZATION REPORT NUMBER(S) AFWL-TR-87-93		
6a. NAME OF PERFORMING ORGANIZATION Applied Technology Associates, Inc.		6b. OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION Air Force Weapons Laboratory		
6c. ADDRESS (City, State, and ZIP Code) Albuquerque, NM 87119			7b ADDRESS (City, State, and ZIP Code) Kirtland Air Force Base, NM 87117-6008		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F29601-86-C-0252		
8c. ADDRESS (City, State, and ZIP Code)			10 SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO 65502F	PROJECT NO 3005	TASK NO AR
			WORK UNIT ACCESSION NO 07		
11 TITLE (Include Security Classification) VAX TO CRAY NASTRAN USER INTERFACE					
12 PERSONAL AUTHOR(S) James, M. R.					
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM Jul 86 TO Jul 87		14 DATE OF REPORT (Year, Month, Day) 1988 October	
				15 PAGE COUNT 78	
16 SUPPLEMENTARY NOTATION					
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	NASTRAN, PATRAN, CRAY, VAX, Silicon Graphics, NASTRAN User Interface NASFILE		
19 ABSTRACT (Continue on reverse if necessary and identify by block number)					
The goal of this program is to overcome the difficulties in establishing fast and effective communication links between certain large computer systems. Previously, file transfer between PATRAN installed on the Silicon Graphics and on the VAX 8700, and NASTRAN installed on the CRAY-1S, was restricted because transferable files between these machines could not be easily produced; resulting in time-consuming and error-prone procedures. This software simplifies such procedures.					
20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS			21 ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL Mr. K. Qassim			22b TELEPHONE (Include Area Code) (505) 844-2229		22c OFFICE SYMBOL ARBC

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

TABLE OF CONTENTS

	page
1. OVERVIEW OF SILICON GRAPHICS/VAX/CRAY AND NASTRAN INTERFACE	3
PREREQUISITES FOR NASTRAN/PATRAN CRAY/VAX MODAL ANIMATION	3
2. SUMMARY TABLE FOR NASTRAN TO PATRAN VAX/CRAY/VAX PROCESSING	6
STEP 0: (SG) DEVELOP THE NASTRAN MODEL	8
STEP 1: (SG) TRANSFER MODEL TO VAX	8
STEP 2: (VAX) TRANSLATE MODEL	8
STEP 3: (CRAY) RETRIEVE MODEL AND RUN NASTRAN	9
STEP 4: (CRAY) RUN NASPAT	10
STEP 5: (VAX) RETRIEVE RESULTS	11
STEP 6: (VAX) TRANSFER FILE FROM VAX TO SILICON GRAPHICS	11
STEP 7: (SG) TRANSLATE RESULTS	12
STEP 8: (SG) RUN PATRAN	12
3. EXAMPLES AND EXTRA FEATURES	14
NASTRAN DMAP INSTRUCTIONS	14
USING THE CRAY ALTER LIBRARY, 'NASALTR'	14
EXAMPLE	15
CONTENTS OF THE NASALTER LIBRARY	16
4. EXAMPLE OF VAX/CRAY/VAX PROCESSING	29
Appendix - MODAL STRAIN ENERGY TABULATION (MSET)	53

1. OVERVIEW OF SILICON GRAPHICS/VAX/CRAY AND NASTRAN INTERFACE

The Air Force Weapons Laboratory (AFWL) has access to both the NASTRAN and PATRAN software systems for structural analysis investigations; however, these two codes are resident on separate computer systems at AFWL. NASTRAN is used to perform the bulk of large structural analysis research and is resident on the CRAY-1S. PATRAN is installed on a Silicon Graphics and allows depictions and animation of modal analysis results. The two software packages are complementary in function, and communication between them can be achieved through the GATEWAY facility. The GATEWAY links the AFWL04 VAX, the Silicon Graphics IRIS, and the CRAY through a common network protocol.

This document describes software resident on the VAX and CRAY computer systems which allows PATRAN animation of NASTRAN calculated modal shapes. This section contains general information which should be reviewed by all potential users. Section 2 outlines the step by step implementation of NASTRAN calculation and PATRAN animation. This approach is designated VAX/CRAY/VAX processing. The VAX/CRAY/VAX approach is appropriate regardless of whether the NASTRAN model was created on the Silicon Graphics using PATRAN or created on the VAX using an editor. Section 3 provides important notes on the use of the existing NASTRAN DMAP alter library, and Section 4 is an example computer session implementing VAX/CRAY/VAX processing as described in Section 2.

PREREQUISITES FOR NASTRAN USE:

The analyst should have:

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
(wall and/or	
Dist	Special
A-1	

- (1) A Silicon Graphics IRIS User account,
- (2) An AFWL04 VAX User account with GATEWAY privileges,
- (3) An AFWL CRAY User account, and
- (4) access to the following files:

on the VAX:

MODAL.COM,

MODAL.EXE

on the CRAY:

NASVAX,

NASGO,

NASGO2,

NASJOB1,

NASJOB2,

NASPRGEX,

MSET,

and NASALTR

It is strongly recommended that the analyst acquaint himself/herself with the operating systems of the VAX, the Silicon Graphics, and CRAY. Many unusual error conditions can occur on all machines, and these conditions are beyond the scope of the present document.*

*For additional help on the VAX or the CRAY, contact the consulting group at 844-8031. Additional help on the Silicon Graphics may be obtained from either the Silicon Graphics User Hotline (1-800-252-0222) or by contacting ATA at 505/247-8371.

2. SUMMARY TABLE FOR NASTRAN TO PATRAN VAX/CRAY/VAX PROCESSING

TABLE I
SUMMARY OF NASTRAN TO PATRAN
VAX/CRAY/VAX PROCESSING

SG:

Step 0: Develop NASTRAN model

Step 1: Transfer model to VAX

VAX:

Step 2: Translate model

CRAY:

Step 3: Retrieve model and
Run NASTRAN

Step 4: Run NASPAT

VAX:

Step 5: Retrieve results

Step 6: Transfer file from VAX
to Silicon Graphics
IRIS

SG:

Step 7: Translate results

Step 8: Run PATRAN

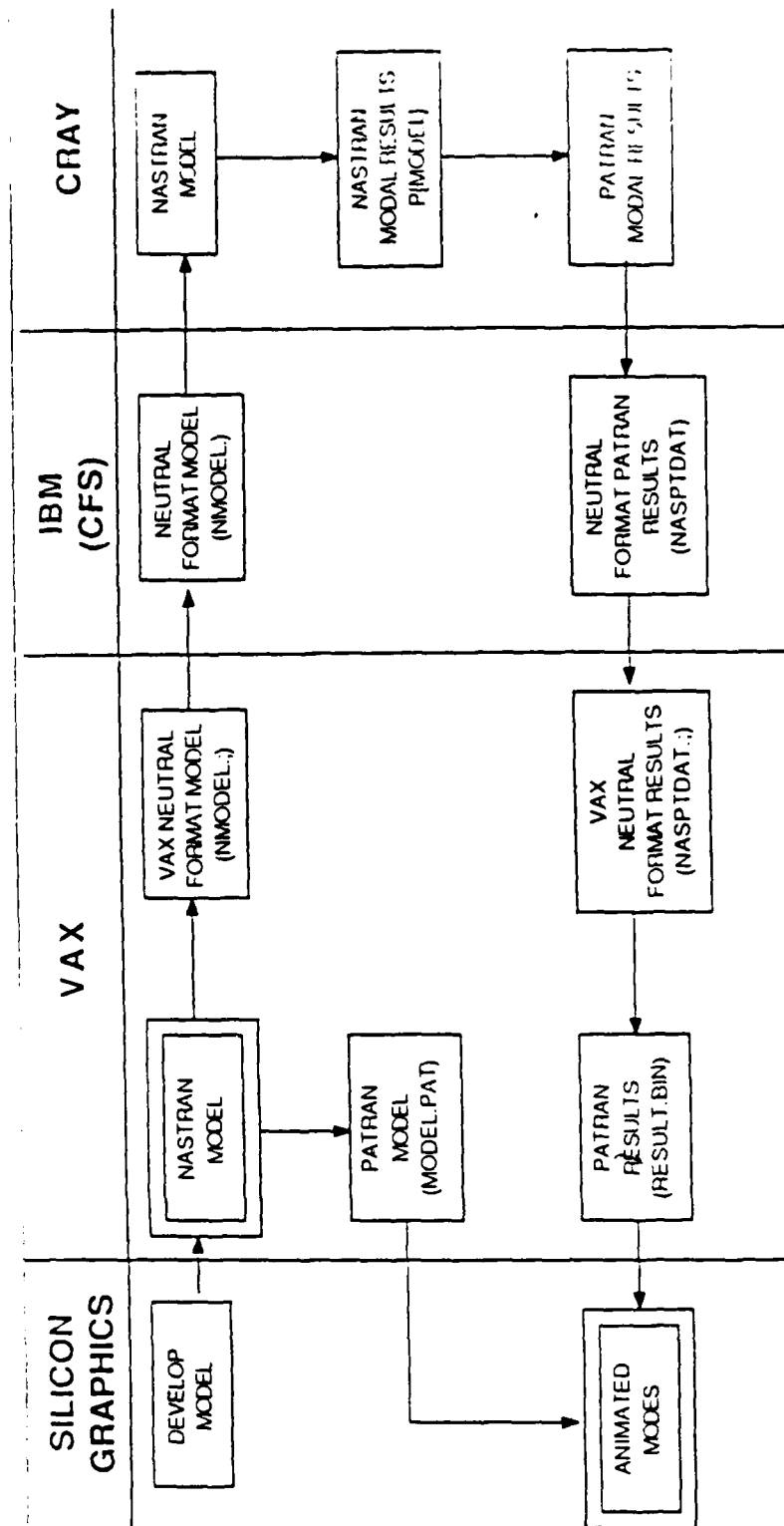


Figure 1. SG/VAX/CRA/VAX Processing Schematic

STEP 0: (SG) DEVELOP THE NASTRAN MODEL: The analyst may use PATRAN on the Silicon Graphics to develop a NASTRAN model. The model should be in upper case once on the VAX; during transport to the CRAY, it will be translated to lower case. Section 3 has some 'special notes' on NASTRAN DMAP sequences.

STEP 1: (SG) TRANSFER MODEL TO VAX: Using a file transfer routine on the Silicon Graphics, transfer the 'PATRAN developed' NASTRAN model to the VAX.

STEP 2: (VAX) TRANSLATE MODEL: The NASTRAN model must be translated into a neutral format for transfer to the CRAY. The neutral format conversion is performed within MODAL.COM as follows:

```
S  @MODAL          --Start the 'Modal' procedure.
    {model file name}  --Supply the file name of the NASTRAN model.

                                A menu of options will appear on the screen.
1                                --Choose option 1 to translate the model into PATRAN
                                format. The menu should re-appear
2                                --Choose option 2 to translate the model into neutral
                                format. The menu should re-appear.
6                                --Choose option 6 to exit.
```

The model should now be stored under CFS and we are ready to log into the CRAY. The user may either log onto the CRAY directly or may reach the CRAY via the GATEWAY. These instructions assume that the GATEWAY will be used.

```

CONNECT          --Initiate GATEWAY bridge to the CRAY.
<CR>             --Hit carriage return or enter user id if different.
<CR>             --Hit carriage return or enter charge code if
                  different.
{CRAY Password}  --Provide correct ICN password.

```

The CRAY should now respond to the user.

If it is necessary to log off of the CRAY, type "**D**" (<CONTROL>D) and control will be returned to the VAX.

If steps 0 through 2 have been completed, and the user has logged onto the CRAY, processing may continue with STEP 3.

STEP 3: (CRAY) RETRIEVE MODEL AND RUN NASTRAN: COSMOS deck 'NASGO' will retrieve the NASTRAN model file from CFS storage and translate it into lower case CRAY ASCII.

```

mass get nasgo    --Retrieve the NASGO COSMOS deck from CFS.
cosmos i=nasgo    --Run NASGO.
{model name}      --Supply NASTRAN model name (up to six characters).
                  The NASTRAN model file will be stored both in local
                  file space and under CFS.

```

NASGO will present a current list of default NASTRAN run variables and prompt the user for instructions. The new user should type "help" for a list of all options.

set --Select an item by item update of current variable values either individually, (e.g., "SET 1 1000000) or all variables. The user should be sure to set item 7 (NASTRAN deck name) and item 2 (user CRAY account number). See also Section 3 of this document.

stop --Accept current variable values and proceed with NASTRAN job processing.

If NASGO encounters no errors, it will submit a job to run NASTRAN using the specified NASTRAN model. This job will appear in the CRAY job queue as "J{model}" where {model} is the name of the NASTRAN model file. The user may verify that this job is in the queue by typing the CTSS command "status."

When the NASTRAN job completes successfully, two files will appear in the user's local file space; these two files are "P{MODEL}" and "NASG02." The appearance of NASG02 in local file space indicates that the user may proceed to Step 4. {MODEL} is the name of the NASTRAN model file.

STEP 4: (CRAY) RUN NASPAT: The COSMOS deck 'NASG02' submits a batch job to run NASPAT to translate NASTRAN results into PATRAN neutral format. It also stores these PATRAN neutral results under CFS for eventual transfer across the GATEWAY to the VAX.

```
cosmos 1=naego2      --Run NASGO2.
{CRAY user number}  --Specify the valid CRAY User ID number.
{job time}          --Specify the maximum NASPAT job time in minutes.
```

NASGO2 will submit a job labeled "JNASPAT." This job will appear on the CRAY job queue when the user types "status." When "JNASPAT" has finished, a file labeled "NASDONE" should appear in local file space. The appearance of this file indicates that CRAY processing is now complete and the user should return to the VAX to run PATRAN.

```
^D                  --Log off from the CRAY (<Control>D).
```

STEP 5: (VAX) RETRIEVE RESULTS: The neutral format PATRAN results file must be retrieved from CFS storage via the GATEWAY. This is accomplished as follows: If the user has not activated the GATEWAY, 'NETON' should be executed as described under STEP 2.

```
S  MASS              --Call the CFS MASS utility.
{VAX Password}      --Supply the correct VMS password.
? GET NASPTDAT       --Request MASS to retrieve the PATRAN results file.
? END                --Exit from MASS.
```

The neutral format PATRAN results file should now be under the name "NASPTDAT.;" in local file space.

STEP 6: (VAX) TRANSFER FILE FROM VAX TO SILICON GRAPHICS: Transfer the file NASPTDAT.; to the silicon graphics for PATRAN execution.

STEP 7: (SG) TRANSLATE RESULTS: NASPAT on the Silicon Graphics may be used to process the neutral format PATRAN results file brought from the CRAY. The file is translated and divided into individual PATRAN binary results files (one for each mode). These files are the individual mode results to be animated by PATRAN.

STEP 8: (SG) RUN PATRAN: PATRAN is used for animation of the mode shapes calculated by NASTRAN. Animation requires both the original model translated into PATRAN format as well as the modal results in PATRAN format (Step 7).

```
$ PATRAN          --Start PATRAN execution.
GO               --Tell PATRAN to get going.
1               --Select new data file.
5               --Select neutral data mode.
2               --Select input model.
(PATRAN model name --Specify neutral model file name.
N               --Neutral input IDs should not be offset.

Y               --Proceed. (Answer "Y" to all queries.)
4               --Select results mode.
1               --Select external data.
1               --Select deformed shape.
(PATRAN result file) --Select result file name. Input the animation file
                    name desired.
5               --Animate mode.
Y               --Type "Y" to repeat animation, "N" to stop.
8               --End.
```

5 --End.

6 --End.

The preceding PATRAN command sequence is given for quick reference only.
Consult the PATRAN manual for all command options.

3. EXAMPLES AND EXTRA FEATURES

NASTRAN DMAP INSTRUCTIONS:

To animate any mode of a structural model, it is necessary to generate a modal displacement file. NASTRAN provides this capability through the use of DMAP instructions. DMAP card sequences instruct the NASTRAN processor to calculate, save, or manipulate data in its internal data arrays. For modal animation purposes, NASTRAN must be instructed to save modal displacement information for each requested mode.

USING THE CRAY ALTER LIBRARY, 'NASALTR:'

A library of standard CRAY DMAP card sequences has been established on the CRAY. This library is named 'NASALTR' and contains sequences for several analysis modes. The CRAY COSMOS decks described in this document access the NASALTR library when constructing the user's NASTRAN job. This utility allows the user to code a NASTRAN deck without including actual DMAP commands. Instead, the user selects the proper DMAP sequence from the NASALTR library at job submit time and the COSMOS deck NASGO writes the appropriate DMAP commands into the user's model immediately preceding the '*cend' card.

WARNING: The user must either choose the appropriate alter deck from the NASALTR library or code in the necessary DMAP commands. If the DMAP sequence is not included when the job is submitted, NASTRAN will run without producing modal displacement results.

EXAMPLE:

The following simple example should be studied and the user's NASTRAN decks tailored accordingly. NASTRAN solution sequence 3 combined with alter deck 'alter1' produces modal displacement results which may be animated by PATRAN on the VAX. The NASTRAN executive deck for model 'x' has been written as

```
*id x
```

```
*sol 3
```

```
*cend
```

When the COSMOS deck NASG01 runs on the CRAY, the user sets the alter deck name (index 10) to "alter1" and the alter library name (index 3 to "nasaltr." Before the actual NASTRAN job is submitted to the CRAY batch queue, it is rewritten to include the specified alter as follows:

```
*id x
```

```
*sol 3
```

```
$ MSA/NASTRAN ALTER
```

```
$ FOR SOLUTION 3 (Normal Modes)
```

```
$ USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES
```

```
ALTER 450 $
```

```
OUTPUT2 OUGV1// -1/11V,N,Z $
```

```
$ CEND
```

\$ DISP = ALL

*cend

CONTENTS OF THE NASALTER LIBRARY:

The following is a reference library of the contents of the NASALTER NASTRAN alter deck library on the CRAY.

----- alter1 -----

\$ MSC/NASTRAN ALTER

\$ FOR SOLUTION 3 (Normal Modes)

\$ USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES

ALTER 450 \$

OUTPUT2 OUGV1// -1/11/V,N,Z \$

\$ CEND

\$ DISP = ALL

----- alter2 -----

\$ MSC/NASTRAN ALTER

\$ FOR SOLUTION 3 (Normal Modes)

\$ FORM THE TRANSFORMATION MATRIX TRANSGB TO

\$ TRANSFORM DISPLACEMENTS INTO GLOBAL RECTANGULAR SYSTEM

ALTER 450 \$

MATMOD CSTM,SIL,BGPD, , , /TRANSGB, /5// -1 \$

MPYAD TRANSGB, UGV/UGVBASIC \$

SDR2 CASECC, , , , EQEXIN, , , , LAMA, , UGVBASIC, , /

```

LAMA,,UGVBASIC,,/
      ,,OUG1VPAT,,,/SOLTYPE/S,N,NOSORT2/V,N,NOCOMP S
OUTPUT2  OUGV1PAT,OES1//-1/11/V,N,ZS

$  CEND

$  DISP  = ALL

$  STRESS = ALL

----- alter3 -----
$  MSC/NASTRAN ALTER
$  FOR SOLUTION 5 (Buckling)
$  USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES
ALTER      142 $
OUTPUT2  OPHIG//-1/11V,N,Z $
$  CEND
$  DISP  = ALL

----- alter4 -----
$  MSC/NASTRAN ALTER
$  FOR SOLUTION 24 (Statics)
$  USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES
ALTER      188 $
OUTPUT2  OUGV1,OES1X//-1/11/V,N,Z S
$  CEND
$  STRESS(VONM) = ALL
$  DISP      = ALL

```

----- alter5 -----

\$ MSC/NASTRAN ALTER
\$ FOR SOLUTION 24 (Statics) - Strain Energy Recovery
\$ USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES
ALTER 190 \$
OUTPUT2 OUGV1,ONRGY1,OESIX// -1/11/V,N,Z \$
\$ CEND
\$ ESE = ALL
\$ DISP = ALL
\$ STRESS = ALL

----- alter6 -----

\$ MSC/NASTRAN ALTER
\$ FOR SOLUTION 24 (Static) - Grid Point Force Recovery
\$ USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES
ALTER 193 \$
OUTPUT2 OUGV1,OGPFB1// -1/11/V,N,Z \$
\$ CEND
\$ DISP = ALL
\$ GPFORCE = ALL

----- alter7 -----

\$ MSC/NASTRAN ALTER
\$ FOR SOLUTION 24 (Statics) - Element Strain Recovery
\$ USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES
ALTER 215 \$

OUTPUT2 OUGV1,OSTR2//-1/11/V,N,Z S

\$ CEND

\$ DISP = ALL

\$ STRAIN(FIBER,VONM) = ALL

----- alter8 -----

\$ MSC/NASTRAN ALTER

\$ FOR SOLUTION 24 (Statics) - Local Displacements to Global System

\$ USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES

\$ FORM THE TRANSFORMATION MATRIX TRANSGB TO

\$ TRANSFORM DISPLACEMENTS INTO GLOBAL RECTANGULAR SYSTEM

ALTER 187 \$

MATMOD CSTM,SIL,BGPDOT,,,/TRANSGB,/5//-1 S

MPYAD TRANSGB,UGV,/UGVBASIC S

SDR2 CASECC,,,EQEXIN,,,,,UGVBASIC,,/

,,OUG1VPAT,,,/STATICS/S,N,NOSORT2/V,N,NOCOMP S

OUTPUT2 OUGV1PAT,OES1X//-1/11/V,N,ZS

\$ CEND

\$ DISP = ALL

\$ STRESS = ALL

----- alter9 -----

\$ MSC/NASTRAN VERSION 63 ALTER

\$ FOR SOLUTION 27 (Direct Transient Analysis)

\$ USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES

ALTER 1 \$

```

OUTPUT2      ,,,,//C,N,-1/C,N,11/V,N,Z $
ALTER        449 $
OUTPUT2      OUGV1,OES1//C,N,0/C,N,11/V,N,Z$
$   CEND
$   DISP    = ALL
$   STRESS  = ALL

```

----- alter10 -----

```

$   MSC/NASTRAN VERSION 63 ALTER
$   FOR SOLUTION 27 (Direct Transient Analysis)
ALTER        1 $
OUTPUT2      ,,,,//C,N,-1/C,N,11/V,N,Z $
ALTER        449 $
MATMOD       CSTM,SIL,BGPD,.,./TRANSGB,/5// -1 $
MPYAD        TRANSGB,UGV,/UGVBASIC $
SDR2         CASEXX,.,.,EQEXIN,.,.,.,UGVBASIC,.,/
              ,.,OUG1VPAT,.,./SOLTYPE/S,N,NOSORT2/V,N,NOCOMP $
OUTPUT2      OUGV1PAT,OES1//C,N,0/C,N,11/V,N,Z$
$   CEND
$   DISP    = ALL
$   STRESS  = ALL

```

----- alter11 -----

```

$   MSC/NASTRAN ALTER
$   FOR SOLUTION 47 (Cyclic Symmetry - Statics)
$   USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES

```

```

ALTER      1 $
OUTPUT2    ,,,,//C,N,-1/C,N,11/V,N,Z $
ALTER      268 $
OUTPUT2    OUGV1,OES1X//O/11/V,N,Z$
$  CEND
$  DISP    = ALL
$  STRESS  = ALL

```

----- alter12 -----

```

$  MSC/NASTRAN ALTER
$  FOR SOLUTION 47 (Cyclic Symmetry - Statics)
$  FORM THE TRANSFORMATION MATRIX TRANSGB TO
$  TRANSFORM DISPLACEMENTS INTO GLOBAL RECTANGULAR COORDINATES
ALTER      1 $
OUTPUT2    ,,,,//C,N,-1/C,N,11/V,N,Z $
ALTER      268 $
MATMOD     CSTM,SIL,BGPD,.,./TRANSGB,/5// -1 $
MPYAD      TRANSGB,FUGV,/UGVBASIC $
SDR2       CASEBK,.,.,EQEXIN,.,.,.,UGVBASIC,./
           ,.,OUGV1PAT,.,./C,N,STATICS/S,N,NOSORT2/V,N,NOCOMP $
OUTPUT2    OUGV1PAT,OES1X//O/11/V,N,Z$
$  CEND
$  DISP    = ALL
$  STRESS  = ALL

```

----- alter13 -----

```

$  MSC/NASTRAN ALTER

```

```

$   FOR SOLUTION 61 (Superelement Statics)
$   USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES
ALTER      1 $
OUTPUT2    ,,,,//C,N,-1/C,N,11/V,N,Z $
ALTER      649 $ (FOR NASTRAN VERSION 64, USE ALTER 664)
OUTPUT2    OUGV1,OES1//O/11/V,N,ZS
$   CEND
$   DISP   = ALL
$   STRESS = ALL

```

----- alter14 -----

```

$   MSC/NASTRAN ALTER
$   FOR SOLUTION 64 (Geometric Nonlinear)
$   USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES
ALTER      1 $
OUTPUT2    ,,,,//C,N,-1/C,N,11/V,N,Z $
ALTER      285 $
OUTPUT2    OUGV1,OES1X//C,N,O/C,N,11/V,N,ZS
$   CEND
$   DISP   = ALL
$   STRESS = ALL

```

----- alter15 -----

```

$   MSC/NASTRAN ALTER
$   FOR SOLUTION 64 (Geometric Nonlinear) - Local Displacements to
Global System
$   FORM THE TRANSFORMATION MATRIX TRANSGB TO

```

\$ TRANSFORM DISPLACEMENTS INTO GLOBAL RECTANGULAR COORDINATES

ALTER 1 \$

OUTPUT2 ,,,,//C,N,-1/C,N,11/V,N,Z \$

ALTER 285 \$

MATMOD FCSTMS,FSILS,FBGPD,.,./TRANSGB,/5// -1 \$

MPYAD TRANSGB,FUGV,/UGVBASIC \$

SDR2 CASEXX,,,FEQEXINS,,,,,UGVBASICS../

,,OUGV1PAT,,,/APP/S,N,NOSORT2/V,N,NOCOMP \$

OUTPUT2 OUGV1,OES1X//C,N,0/C,N,11/V,N,ZS

\$ CEND

\$ DISP = ALL

\$ STRESS = ALL

----- alter16 -----

\$ MSC/NASTRAN ALTER

\$ FOR SOLUTION 66 (Material Nonlinear)

\$ USE ONLY IF DISPLACEMENTS ARE IN GLOBAL COORDINATES

ALTER 1 \$

OUTPUT2 ,,,,//C,N,-1/C,N,11/V,N,Z \$

ALTER 940 \$ (FOR NASTRAN VERSION 64, USE ALTER 662)

OUTPUT2 OUGV1,OES1//O/C,N,11/V,N,ZS

\$ CEND

\$ DISP = ALL

\$ STRESS = ALL

----- alter17 -----

\$ MSC/NASTRAN ALTER

```

$   FOR SOLUTION 66 (Material Nonlinear) - Local Displacements to
Global System

$   FORM THE TRANSFORMATION MATRIX TRANSGB TO

$   TRANSFORM DISPLACEMENTS INTO GLOBAL RECTANGULAR COORDINATES

ALTER      1 $

OUTPUT2     ,,,,//C,N,-1/C,N,11/V,N,Z $

ALTER      940 $

MATMOD      CSTMS,SILS,BGPDTS,,,/TRANSGB,/5// -1 $

MPYAD       TRANSGB,UGV,/UGVBASIC $

SDR2        CASEDR,,,,EQEXIN,,,,PJ1,,UGVBASIC,,/
            ,,,OUGV1PAT,,,/C,N,STATICS/S,N,NOSORT2/V,N,NOCOMP $

OUTPUT2     OUGV1PAT,,,/OESL1//C,N,0/C,N,11/V,N,ZS

$   CEND

$   DISP    = ALL

$   STRESS  = ALL

```

```

----- alter18 -----

$   MSC/NASTRAN ALTER

$   FOR SOLUTION 77 (Cyclic Symmetry Buckling)

$

$   PRE-STRESS STATE

ALTER      1 $

OUTPUT2     ,,,,//C,N,-1/C,N,11/V,N,Z $

ALTER      136 $

MATMOD      CSTMS,SILS,BGPD.,,,,/TRANSGB,/5// -1 $

MPYAD       TRANSGB,UGV,/UGVBASIC $

SDR2        CASECC,,,,EQEXIN,,,,,PHIGBASIC,./

```

```

      , , OUGV1PHIG. , , /SOLTYPE/S,N,NOSORT2/V,N,NOCOMP $
EQUIV      OBES1,OES1PAT/ADDPDA $
OUTPUT2    OUGV1PHIG,OES1PAT/C,N,O/C,N,11/V,N,Z $
$
$  CEND
$  DISP    = ALL
$  STRESS  = ALL

```

```

----- alter19 -----
$  MSC/NASTRAN ALTER
$  FOR SOLUTION 89 (Superelement Transient Heat Transfer)
ALTER      1      $
OUTPUT2    , , , , /C,N, -1/C,N,11/V,N,Z $
ALTER      252    $ (FOR NASTRAN VERSION 64, USE ALTER 966)
OUTPUT2    OUGV1,OES1//O11/V,N,Z$
$  CEND

```

```

----- alter20 -----
$  ----- MSET -----
$  CSA Engineering, Inc.
$
$  Alter for MSC/NASTRAN version 65. SOL 63
$  Writes strain energies on OUTPUT2 file for post-processing
$  along with mass info in OUTPUT4 format.
$  WG, Dec 85
$  Rev Mar 86 V65
$  Rev Apr 86 -- handle alternate coord sys WG

```

\$ Rev May 86 -- more DBFETCHs so it works with restart WG
 \$ Rev Sep 86 -- PARAM.NOSEPOST
 \$
 \$ Provide file assignments for units 11 and 12 as follows (VAX):
 \$
 \$ \$ ASSIGN [directory]job.OUS FOR011 (strain energy)
 \$ \$ ASSIGN [directory]job.MAS FOR012 (weight)
 \$ \$ ASSIGN [WG.DMAP]CASECC.OU2 FOR015 (read only)
 \$
 \$ Unit numbers may be changed by
 \$
 \$ PARAM ESEUNIT xx (default 11)
 \$ PARAM WTUNIT xx (default 12)
 \$
 \$ PARAM NOSEPOST -1 will skip all calculations in this alter
 \$
 \$ If ESE= is included in the Case Control deck, then strain energies
 \$ will be printed in the usual manner. All strain energies are
 \$ written
 \$ to the auxiliary file irrespective of any ESE request.
 \$
 \$ This alter deals only with the residual structure. Upstream
 \$ superelement energies are not considered.
 \$
 \$ ALTER 1120 \$ V65
 ALTER 1081 \$ V63
 PARAM //NOP/V,Y,NOSEPOST=1 \$

```

COND      NOSEPOST,NOSEPOST $
$
$ Generate grid point mass vector
$
DBFETCH  /MGG,,,,/MODEL/0/0 $
DBFETCH  /BGPPTS,EQEXINS,CSTMS,,/0/0 $
DIAGONAL MGG/MGGDIAGL $
VECPLOT  ,,BGPPTS,EQEXINS,CSTMS,,/DGX6T/-1//4 $
MATGEN   ,/PMASS/6/6/1/5 $
PARTN    DGX6T,,PMASS/DGX1T,,,/1 $
TRNSP    DGX1T/DGX1 $
MPYAD     MGG,DGX1,/MASS $
PARAMR   //DIV/V,N,WTMASSI/1./V,Y,WTMASS $
PARAMR   //COMPLEX//WTMASSI/0./V,N,WTMASSC $
ADD       MASS,/WEIGHTL/WTMASSC $
$
$ Generate strain energy data block
$
INPUTT2  /DUMCASE,,,,/15/ $
DBFETCH  /UGVS,KELM,KDICT,ECTS,GPECT/0/0 $
DBFETCH  /PG,QGS,SILS,GPLS,VELEM/0/0 $
GPFDR
DUMCASE,UGVS,KELM,KDICT,ECTS,EQEXINS,GPECT,PG,QGS,BGPPTS,SILS,
      CSTMS,VELEM/ONRGY1,OGPFB1/APP1/0.0 $
VECPLOT  WEIGHTL,BGPPTS,EQEXINS,CSTMS,CASECC,/WEIGHT/0/0/1 $
VECPLOT  MGGDIAGL,BGPPTS,EQEXINS,CSTMS,CASECC,/MGGDIAG/0/0/1 $
VECPLOT  UGVS,BGPPTS,EQEXINS,CSTMS,CASECC,/UGVSB/0/0/1 $

```

DBFETCH /LAMA,,,,/O/O S

OUTPUT2 LAMA,ONRGY1,BGPDTS,GPLS, //O/V,Y,ESEUNIT=11 S

OUTPUT4 WEIGHT,MGGDIAG,UGVSB, //O/V,Y,WTUNIT=12 S

LABEL NOSEPOST S

4. EXAMPLE OF VAX/CRAY/VAX PROCESSING

The following pages represent a log of an actual computer session which produced animated modal results. This example follows the outline of VAX/CRAY/VAX processing presented in Section 2. For the sake of brevity and legibility, the following conventions have been adopted in the log:

(1) All user input is underlined to distinguish it from computer responses.

(2) On some occasions, actual user input is not echoed in the log. These blank inputs are either passwords which do not appear on the screen during processing single carriage returns <CR>.

THE EXAMPLE MODEL:

Three PATRAN renderings of the example model are appended to the end of the computer session log. The wire mesh and first hidden line drawing represent the undeformed model, while the second hidden line drawing is a highly exaggerated depiction of a calculated structural mode.

For moderately complex models such as this one, the benefits of modal animation become clear. Without animation, careful comparison of the deformed and undeformed node locations may not easily reveal all of the modal analysis information present in these plots. However, with modal animation, the same information is conveyed in a few seconds rather than a few hours.

If double quotes are shown during data entry for program NASPRGEX in the printout, then a carriage return was typed.

neton

.....
If there are any problems, notify Brian Ridout at 844-1654.
MAIL to: XNET2::RIDOUT.
.....

icn password: _____
000 znumber=001995 class=U charge=00003434 distribution=
000 You are now authorized

@modal

>> This is the DCL procedure "MODAL". <<
>> "MODAL" performs all data and model <<
>> translations required for NASTRAN/ <<
>> PATRAN CRAY/VAX processing. <<

>> APPLIED TECHNOLOGY ASSOCIATES <<
>> December, 1985 <<

What is the name of the NASTRAN model file?: plate.dat

>>> CURRENT MODEL IS "PLATE.DAT" <<<

OPTION

- 1 NASTRAN MODEL ---> PATRAN MODEL
Translate NASTRAN model into PATRAN format.
- 2 VAX MODEL ---> CRAY MODEL
Translate model into machine-independent format.

- 3 CRAY MODEL ---> VAX MODEL
 Translate model from machine-independent format.
- 4 CRAY RESULTS ---> VAX RESULTS
 Translate PATRAN result file to VAX format.
- 5 SELECT NEW MODEL
- 6 EXIT

OPTION: 2

RUNNING STEXT

%DELETE-I-FILDEL, DUAL:[JAMES.NASTRAN]NMODEL.;1 deleted (12 blocks)

>> The model will now be put on mass <<
>> storage as NMODEL. <<

%DCL-I-SUPERSEDE, previous value of SYSSINPUT has been superseded

VMS LOGIN PASSWORD:

000 87/07/09 11:16:17.453 STORE NMODEL.:/001995/NMODEL.

001 (115620B BITS)

>>> CURRENT MODEL IS "PLATE.DAT" <<<

OPTION

- 1 NASTRAN MODEL ---> PATRAN MODEL
 Translate NASTRAN model into PATRAN format.
- 2 VAX MODEL ---> CRAY MODEL
 Translate model into machine-independent format.
- 3 CRAY MODEL ---> VAX MODEL
 Translate model from machine-independent format.

- 4 CRAY RESULTS ---> VAX RESULTS
Translate PATRAN result file to VAX format.
- 5 SELECT NEW MODEL
- 6 EXIT

OPTION: 6

ATA \$ mass

VMS LOGIN PASSWORD: _____

? LIST

NODE NAME: 001995

DESCENDANTS:

JCLNAST
JCLPREP
NASTEXE
RATSEXE
NASTPLOT
PLOTEXE
PLTPREP
NPATEXE
NASGO
NASGO1
NASGO2
NASPRGEX
NASALTR
NASPRG
NASPAT
NASPATEX
NASVAX
NASJOB1
NASJOB2
APC
CPAT

TAPE999
 TAPE909
 TAPE199
 PLTX
 MHDCBE
 MHDCB1
 OUTPUT DIR
 NASALTRS
 NASMODEL
 NASGOLD2
 NASGOLD1
 NMODEL.
 MSET.FOR
 MSET
 PLATE
 TAPE7
 DBPLATE DIR
 TAPE6
 MPLATE
 OPLATE
 APLATE

? END

ATA \$ connect

CONNECT Version 4.1 12/86

USE LOWER CASE FOR CTSS

DEFAULTS MAY BE USED FOR USER NUMBER AND CHARGE CODE

THE DEFAULT WILL BE THE VALUES IN THE DP

AUTHORIZATION FILE

(SHOWN BY NETON, WHICH MUST BE RUN BEFORE CONNECT)

CTSS USER NUMBER (DEFAULT - NETON USER NUMBER): _____

CTSS CHARGE CODE (DEFAULT - NETON CHARGE CODE): _____

ICN PASSWORD: _____

V b07f 0180.420 ACTIVE A
NIL

files

6014 rw logmlgy3
5526 rw logtlgy3
2744 re nasgo
107615 re nasprgex
1011 rw plate
73 rw tape7

all done

cosmos i=nasgo

10:57:13 000:00.003 v07/09/87 AFWLCC cosmos 2.4 ar3 001995 mic *** +cosmva0
*** nasgo james m *** page 1

```
1 */  >> THIS IS COSMOS DECK "NASGO".  <<
2 */  >> IT IS DESIGNED TO RETRIEVE A  <<
3 */  >> NASTRAN MODEL WHICH HAS BEEN  <<
4 */  >> SENT TO CFS FROM THE VAX.     <<
5 */
6 */  >> APPLIED TECHNOLOGY ASSOCIATES <<
7 */  >>      December,1985           <<
8 */
9 */
10 */  >> GET THE MODEL FILE (IN STEXT FORMAT) <<
11 */  >> FROM MASS STORAGE FILE "NMODEL."  <<
12 */
```

10:57:15 000:00.165 13 *mass get nmodel.
000 87/07/09 10:57:46.852 get nmodel.:/001995/nmodel.
001 (115620b bits) 87/07/07 09:45:42.837
10:57:19 000:00.440 all done

```

10:57:19 000:00.442    14 *files nmodel.
                        error in input line.
10:57:21 000:00.544      all done
10:57:21 000:00.547    15 *if lastmsg .has. "none" then go to nomodel
16 */
17 */    >> ASK USER FOR THE MODEL NAME <<
18 */
10:57:21 000:00.557    19 *let query=\lf\_ "What is the NASTRAN model name?    {up
                        to six characters}"
                        plate
10:57:21 000:00.565    20 *let i=getmsg(\query\)
10:57:26 000:00.614    21 *let modname=msg
22 */
23 */    >> TRANSLATE THE MODEL INTO CRAY ASCII <<
24 */    >> FORMAT AND CONVERT TO LOWER CASE.    <<
25 */
10:57:26 000:00.622    26 *ntext nmodel. ascmodel
10:57:28 000:00.715      all done
27 */
28 */    >> CONVERT THE NASTRAN DECK TO LOWER CASE    <<
29 */
10:57:28 000:00.721    30 *trans i=(ascmodel,cray),o=(\modname\,cray),lc
10:57:32 000:00.807      all done
31 */
32 */    >>STORE THE MODEL UNDER CFS.                                <<
33 */
10:57:32 000:00.813    34 *mass store \modname\
000 87/07/09 10:58:03.949 store plate:/001995/plate
001 (76400b bits)
10:57:37 000:01.086      all done
10:57:37 000:01.088    35 *destroy nmodel. ascmodel
10:57:38 000:01.140      all done
36 */
37 */    -----
38 */    >> "NASGO" IS FINISHED.                                <<

```

39 */ >> THE NASTRAN MODEL IS UNDER CFS <<

10:57:38 000:01.148 v07/09/87 AFWLCC cosmos 2.4 ar3 001995 mic *** .cosmva0 ***
nasgo james m *** page 2

40 */ >> STORAGE AND IN LOCAL FILE SPACE. <<

41 */ >> READY TO RUN "NASG01"! <<

42 */ -----

43 */

10:57:39 000:01.198 44 *go to endl

10:57:39 000:01.205 57 *endl:

58 */ >> THIS IS COSMOS DECK "NASG01". <<

59 */ >> IT IS DESIGNED TO EXECUTE THE FIRST <<

60 */ >> PHASE OF NASTRAN/PATRAN VAX/CRAY <<

61 */ >> MODAL ANALYSIS. "NASG01" WRITES <<

62 */ >> AND SUBMITS A NASTRAN JOB. <<

63 */

64 */ >> APPLIED TECHNOLOGY ASSOCIATES <<

65 */ >> November, 1985 <<

66 */

67 */

10:57:40 000:01.261 68 *select messages=short

10:57:40 000:01.265 69 *let iero="no"

70 */

71 */ >> SEARCH FOR THE NASPROG EXECUTABLE <<

72 */

10:57:40 000:01.273 73 *let errorfile="nasprgex"

10:57:40 000:01.279 74 *files nasprgex

none

10:57:42 000:01.391 all done

10:57:42 000:01.393 75 *if lastmsg .has. "none" then mass get nasprgex

000 87/07/09 10:58:13.217 get nasprgex/root:/001995/nasprgex

001 (10761500b bits) 87/06/11 11:23:26.993

10:57:47 000:01.673 all done

10:57:47 000:01.676 76 *files nasprgex

107615 re nasprgex

```

10:57:49 000:01.780      all done
10:57:49 000:01.782    77 *if lastmsg .has. "none" then go to missingfile
                        78 */
                        79 */  >> RUN NASPROG <<
                        80 */
10:57:49 000:01.791    81 *select extralines=tty
10:57:49 000:01.795    82 *nasprgex

```

No tape7 found...

INDEX DESCRIPTIONCURRENT VALUE

10:58:02 000:02.689 v07/09/87 AFWLCC cosmos 2.4 ar3 001995 mic *** +cosmva0 ***
nasgo james m *** page 3

1	Memory Coresize	{ 6 digits }	300000
2	User ID number	{ 6 digits }	000000
3	Alter Library Name		nasaltr
4	Printer Device		0
5	Checkpoint Restart Name		0
6	Database Restart Name		0
7	NASTRAN deck name		
8	Execution Time Limit {minutes}		10
9	Job Priority { 1.00 to 1.99 }		1.00
10	Alter Deck Name		0
11	Class	{ a, b, or c }	a
12	MSET run?	{0=No, 1=Yes}	0

**** Option? Type "help" for help

?

+ "set"

> Memory Coresize { 6 digits } CURRENTLY = 300000 <CR> to keep

?

+ "100000"

> User ID number { 6 digits } CURRENTLY = 000000 <CR> to keep

?

+ "001995"

> Alter Library Name CURRENTLY = nasaltr <CR> to keep

?

+ """

> Printer Device CURRENTLY = 0 <CR> to keep

?

+ """

> Checkpoint Restart Name CURRENTLY = 0 <CR> to keep

?

+ """

> Database Restart Name CURRENTLY = 0 <CR> to keep

?

+ """

> NASTRAN deck name CURRENTLY = <CR> to keep

?

+ "plate"

> Execution Time Limit {minutes} CURRENTLY = 10 <CR> to keep

?

+ "2"

> Job Priority { 1.00 to 1.99 } CURRENTLY = 1.00 <CR> to keep

?

+ """

> Alter Deck Name CURRENTLY = 0 <CR> to keep

?

+ """

> Class { a, b, or c } CURRENTLY = a <CR> to keep

?

+ """

> MSET run? {0=No, 1=Yes} CURRENTLY = 0 <CR> to keep

?

+ "1"

11:01:07 000:07.153 v07/09/87 AFWLCC cosmos 2.4 ar3 001995 mic *** +cosmva0 ***
nasgo james m *** page 4

INDEX	DESCRIPTION	CURRENT VALUE
1	Memory Coresize { 6 digits }	100000
2	User ID number { 6 digits }	001995
3	Alter Library Name	nasaltr
4	Printer Device	0
5	Checkpoint Restart Name	0
6	Database Restart Name	0
7	NASTRAN deck name	plate
8	Execution Time Limit {minutes}	2
9	Job Priority { 1.00 to 1.99 }	1.00
10	Alter Deck Name	0
11	Class { a, b, or c }	a
12	MSET run? {0=No, 1=Yes}	1

**** Option? Type "help" for help
?
+ "stop"

>>>>> TAPE7 has been created
stop
nasprgex ctss time 2.116 seconds
cpu= .061 i/o= 9.425 memory time= .183
11:01:37 000:08.809 all done
11:01:37 000:08.812 83 *select extralines=none
84 */
85 */
86 */ >> DETERMINE WHETHER THIS IS A RATS RUN <<
87 */

```

11:01:37 000:08.821      88 *files tape8
                           none
11:01:39 000:08.937      all done
11:01:39 000:08.939      89 *if lastmsg .has. "none" then go to normal
11:01:39 000:08.954      112 *normal:
                           113 */
                           114 */ >> THIS IS A NORMAL NASTRAN RUN <<
                           115 */
11:01:39 000:08.959      116 *let input7="tape7"
                           117 */
                           118 */ >> READ IN THE PROCESSING PARAMETERS FROM <<
                           119 */ >> THE TAPE7 FILE GENERATED BY NASPROGE. <<
                           120 */
11:01:39 000:08.970      121 *qed \input7\\lf\11\\lf\end
                           12 lines. (a) tape7
                           100000      Memory Coresize      { 6 digits }
11:01:42 000:09.131      all done
11:01:42 000:09.134      122 *if lastmsg .has. "abort" then go to aborted
11:01:42 000:09.140      123 *let temp=rplcc(lastmsg," ")
11:01:42 000:09.148      124 *let coresize=getsym(temp,1)
11:01:42 000:09.154      125 *qed \input7\\lf\21\\lf\end

11:01:43 000:09.214 v07/09/87 AFWLCC cosmos 2.4 ar3 001995 mic *** +cosmva0 ***
nasgo james m *** page 5

                           12 lines. (a) tape7
                           001995      User ID number      { 6 digits }
11:01:45 000:09.319      all done
11:01:45 000:09.322      126 *let temp=rplcc(lastmsg," ")
11:01:45 000:09.329      127 *let userid=getsym(temp,1)
11:01:45 000:09.336      128 *qed \input7\\lf\31\\lf\end
                           12 lines. (a) tape7
                           nasaltr      Alter Library Name
11:01:48 000:09.497      all done
11:01:48 000:09.499      129 *let temp=rplcc(lastmsg," ")
11:01:48 000:09.507      130 *let alterlibrary=getsym(temp,1)

```

```

11:01:48 000:09.513 131 *qed \input7\\lf\41\\lf\end
                        12 lines. (a) tape7
                        0          Printer Device
11:01:51 000:09.675      all done
11:01:51 000:09.677 132 *let temp=rplcc(lastmsg," ")
11:01:51 000:09.685 133 *let printerlist=getsym(temp,1)
11:01:51 000:09.691      134 *if printerlist .eq. "printer" then let
printerlist=" "
11:01:51 000:09.698 135 *qed \input7\\lf\51\\lf\end
                        12 lines. (a) tape7
                        0          Checkpoint Restart Name
11:01:53 000:09.860      all done
11:01:53 000:09.862 136 *let temp=rplcc(lastmsg," ")
11:01:53 000:09.870 137 *let ckpt="no"
11:01:53 000:09.875 138 *let rstddb="no"
11:01:54 000:09.881 139 *let temp1=getsym(temp,1)
11:01:54 000:09.887 140 *if temp1 .ne. "0" then let ckpt="yes"
11:01:54 000:09.894 141 *if ckpt .eq. "yes" then go to skip
142 */
11:01:54 000:09.901 143 *qed \input7\\lf\61\\lf\end
                        12 lines. (a) tape7
                        0          Database Restart Name
11:01:57 000:10.063      all done
11:01:57 000:10.065 144 *let temp=rplcc(lastmsg," ")
11:01:57 000:10.073 145 *let temp1=getsym(temp,1)
11:01:57 000:10.079 146 *if temp1 .ne. "0" then let rstddb="yes"
147 */
11:01:57 000:10.086 148 *skip:
11:01:57 000:10.088 149 *let restartid=getsym(temp1,1)
11:01:57 000:10.095 150 *if temp1 .eq. "0" then let restartid=" "
11:01:57 000:10.104 151 *qed \input7\\lf\71\\lf\end
                        12 lines. (a) tape7
                        plate      NASTRAN deck name
11:02:00 000:10.266      all done
11:02:00 000:10.269 152 *let temp=rplcc(lastmsg," ")
11:02:00 000:10.276 153 *let nasdeck=getsym(temp,1)

```

```

11:02:00 000:10.283      154 *qed \input7\\lf\81\\lf\end
                           12 lines. (a) tape7
                           2           Execution Time Limit {minutes}

11:02:02 000:10.445      all done

11:02:02 000:10.447      155 *let temp=rplcc(lastmsg," ")

11:02:02 000:10.455 v07/09/87 AFWLCC cosmos 2.4 ar3 001995 mic *** +cosmva0 ***
nasgo james m *** page 6

11:02:02 000:10.459      156 *let maxtime=getsym(temp,1)

11:02:02 000:10.465      157 *qed \input7\\lf\91\\lf\end
                           12 lines. (a) tape7
                           1.00       Job Priority { 1.00 to 1.99 }

11:02:05 000:10.628      all done

11:02:05 000:10.630      158 *let temp=rplcc(lastmsg," ")

11:02:05 000:10.638      159 *let xpriority=getsym(temp,1)

11:02:05 000:10.645      160 *qed \input7\\lf\101\\lf\end
                           12 lines. (a) tape7
                           0           Alter Deck Name

11:02:07 000:10.806      all done

11:02:07 000:10.809      161 *let temp=rplcc(lastmsg," ")

11:02:07 000:10.817      162 *let altid=getsym(temp,1)

11:02:07 000:10.823      163 *let altr="no"

11:02:07 000:10.829      164 *if altid .ne. "0" then let altr="yes"

11:02:07 000:10.836      165 *qed \input7\\lf\111\\lf\end
                           12 lines. (a) tape7
                           a           Class           { a, b, or c }

11:02:10 000:10.998      all done

11:02:10 000:11.001      166 *let temp=rplcc(lastmsg," ")

11:02:10 000:11.009      167 *let xclass=getsym(temp,1)

11:02:10 000:11.016      168 *qed \input7\\lf\121\\lf\end
                           12 lines. (a) tape7
                           1           MSET run?       {0=No, 1=Yes}

11:02:13 000:11.178      all done

11:02:13 000:11.180      169 *let temp=rplcc(lastmsg," ")

11:02:13 000:11.188      170 *let misset=getsym(temp,1)

```

```

11:02:13 000:11.194      171 *let mmset="no"
11:02:13 000:11.200      172 *if mset .ne. "0" then let mmset="yes"
                               173 */
                               174 */ >> IF "NASJOB1" IS NOT IN LOCAL SPACE, <<
                               175 */ >> LOOK FOR IT UNDER MASS STORAGE.    <<
                               176 */

11:02:13 000:11.214      177 *let errorfile="nasjob1"
11:02:13 000:11.220      178 *let jclempty="nasjob1"
11:02:13 000:11.227      179 *files \jclempty\
                               none
11:02:15 000:11.339          all done
11:02:15 000:11.341      180 *if lastmsg .has. "none" then mass get \jclempty\
                               000          87/07/09          11:02:46.518          get
nasjob1:/001995/nasjob1
                               001 (246300b bits) 87/07/09 08:47:47.822

11:02:19 000:11.621          all done
11:02:19 000:11.624      181 *files \jclempty\
                               2463 re nasjob1

11:02:21 000:11.728          all done
11:02:21 000:11.730      182 *if lastmsg .has. "none" then go to missingfile
                               183 */
                               184 */ >> SET UP PARAMETERS FOR USE IN "NASJOB1" <<
                               185 */

11:02:21 000:11.741      186 *let printerlist='''_printerlist\_'''
11:02:21 000:11.751      187 *let userid='''_userid\_'''

11:02:21 000:11.761 v07/09/87 AFWLCC cosmos 2.4 ar3 001995 mic *** +cosmva0 ***
nasgo james m *** page 7

11:02:21 000:11.765      188 *let coresize='''_coresize\_'''
11:02:21 000:11.775      189 *let copyinput="no"
11:02:21 000:11.781      190 *let jobname="j"_nasdeck\
11:02:21 000:11.790      191 *let mset='''_mmset\_'''
                               192 */
                               193 */ >>IF NASTRAN DECK IS NOT IN LOCAL FILE SPACE.<<
                               194 */ >> LOOK FOR IT UNDER MASS STORAGE.    <<

```

```

11:02:22 000:11.847 195 */
11:02:22 000:11.853 196 *let errorfile=\nasdeck\
197 *files \nasdeck\
764 rw plate
11:02:24 000:11.961 all done
11:02:24 000:11.964 198 *if lastmsg .has. "none" then mass get \nasdeck\
11:02:24 000:11.971 199 *files \nasdeck\
764 rw plate
11:02:26 000:12.075 all done
11:02:26 000:12.077 200 *if lastmsg .has. "none" then go to missingfile
201 */
11:02:26 000:12.085 202 *if \ckpt\ .ne. "yes" then go to shipit
11:02:26 000:12.101 219 *shipit:
220 */
221 */ >> DESTROY ANY OLD VERSIONS OF THE JOB CONTROL<<
222 */ >> FILE FOR THIS NASTRAN DECK. <<
223 */
11:02:26 000:12.107 224 *destroy \jobname\
not found:
jplate
11:02:28 000:12.211 all done
225 */
226 */ >> DETERMINE WHETHER THE NASTRAN DECK BEGINS <<
227 */ >> WITH A "*DECK" CARD. IF NOT, INSERT ONE. <<
228 */
11:02:28 000:12.219 229 *qed \nasdeck\\lf\l1\lf\end
157 lines. (a) plate
id weight,test
11:02:31 000:12.401 all done
11:02:31 000:12.404 230 *let temp=rplcc(lastmsg," ")
11:02:31 000:12.412 231 *let firstword=getsym(temp,1)
11:02:31 000:12.419 232 *if firstword .eq. "*deck" then go to skip2
233 */
11:02:31 000:12.427 234 *trixgl o(\nasdeck\\lf\b11\lf\*deck
\nasdeck\\lf\.\end
%missing end \, will supply one.

```

157 lines (80s)

%trying to end controllee end

11:02:34 000:12.718 all done

235 */

11:02:34 000:12.722 236 *skip2:

11:02:34 000:12.723 237 *select messages=short

238 */

239 */ >> DETERMINE WHETHER ALTER LINES EXIST <<

11:02:34 000:12.731 v07/09/87 AFWLCC cosmos 2.4 ar3 001995 mic *** +cosmva0 ***
nasgo james m *** page 8

240 */ >> OR ARE NEEDED FOR THIS RUN. <<

241 */ >>(IS THIS A CHECKPOINT OR ALTER LIBRARY RUN?)<<

242 */

11:02:35 000:12.779 243 * if (ckpt .ne. "yes") .and. (altr .ne. "yes") then
go to noalters

11:02:35 000:12.838 353 *noalters:

354 */

355 */ >> MERGE THE NASTRAN DECK INTO THE <<

356 */ >> JOB CONTROL FILE (NASJOB1). <<

357 */

11:02:35 000:12.845 358 *trixgl \lf\mf(\nasdeck\ \jclempty\,\jobname\)
ok. jplate merged

11:02:38 000:13.027 all done

11:02:38 000:13.030 359 *trixgl o(\jobname\)\lf\fp\lf*select\lf\l%

% %

589 lines (80s)

176 *select message=medium,savefiles=none

360 tp1,j\lf*let coresize=\lf\rl%

182 *let coresize="300000"

361 tp1,j\lf*let userid=\lf\rl%

184 *let userid="000000"

```

362 tp1,j\lf\*let printerlist=\lf\rl%
183 *let printerlist="mfaco"
11:02:49 000:13.938 all done
363 */
364 */ >> DESTROY THE MODIFIED NASTRAN DECK <<
365 */ >> IF IT WAS CREATED, AND RESTORE THE <<
366 */ >> ORIGINAL UNALTERED COPY. <<
367 */
11:04:05 000:13.988 368 *if \copyinput\ .eq. "no" then go to noextrainput
11:04:05 000:13.998 372 *noextrainput:
373 */
11:04:05 000:14.001 374 *if \ckpt\ .eq. "no" then go to dbrcheck
11:04:06 000:14.054 382 *dbrcheck:
11:04:06 000:14.056 383 *if \rstdb\ .eq. "no" then go to endjclprp
11:04:06 000:14.083 427 *endjclprp:
428 */
429 */ >> IF NO ERRORS HAVE BEEN ENCOUNTERED, <<
430 */ >> SUBMIT THE JOB. <<
431 */
11:04:06 000:14.089 432 *mass store tape7
000 87/07/09 11:04:37.649 store tape7:/001995/tape7
001 (7300b bits)
11:04:11 000:14.400 all done
11:04:11 000:14.402 433 *destroy alwith. %%% %
11:04:12 000:14.454 all done
11:04:12 000:14.456 434 *if \iero\ .eq. "yes" then go to flagerror
435 */
11:04:12 000:14.466 436 *submit
i=\jobname\,t=\maxtime\,p=\xpriority\,class=\xclass\ end
jplate was submitted.

11:04:16 000:14.669 v07/09/87 AFWLCC cosmos 2.4 ar3 001995 mic *** +cosmva0 ***
nasgo james m *** page 9

11:04:16 000:14.672 all done

```

```

437 */
438 */ !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
439 */ >> THE JOB HAS BEEN SUBMITTED <<
440 */ !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
441 */
11:04:17 000:14.722 442 *go to theend
11:04:17 000:14.727 450 *theend:
11:04:17 000:14.728 451 *destroy tape8 \jobname\ nasjob1 ratsexe data
                        not found:
                        tape8  ratsexe      data
11:04:19 000:14.835      all done
11:04:19 000:14.837 452 *aborted:
11:04:19 000:14.838      > end cosmos run *****

```

all done

files

```

6014 rw logmlgy3
2744 re nasgo
107615 re nasprgex
1011 rw plate
73 rw tape7

```

all done

status j=jplate

division ar3/ata(999917,999918,999919,999920,999922,999923)

job	user	date	time	state	r	limit	d	tid	c	pri
jplate	001995	07/09	08:43:21	ran	3	.72	00		a	1.00
jplate	001995	07/09	11:13:23	ran	3	.77	00		a	1.00

all done

files

```

6014 rw logmlgy3
5526 rw logtlgy3

```

2744 re nasgo
107615 re nasprgex
1011 rw plate
73 rw tape7

all done
mass list

node name: 001995

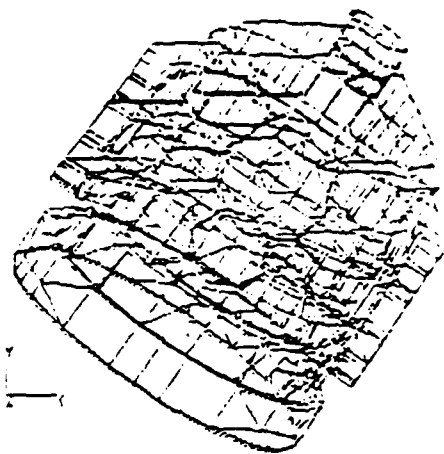
descendants:

jclnast
jclprep
nastexe
ratsexex
nastplot
plotexe
pltprep
npatexe
nasgo
nasgol
nasgo2
nasprgex
nasaltr
nasprg
naspat
naspatex
nasvax
nasjob1
nasjob2
apc
cpat
tape999
tape909
tapel99
pltx
mhdcbx

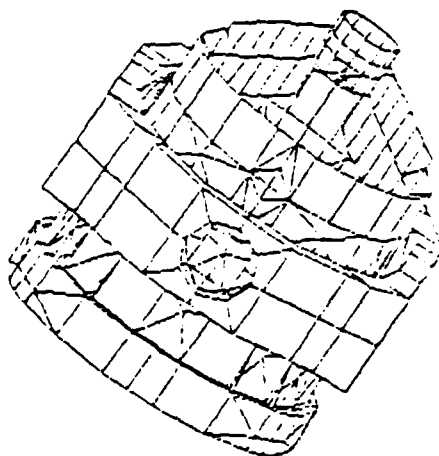
mhdcbl	
output	dir
nasaltrs	
nasmodel	
nasgold2	
nasgold1	
nmodel.	
mset.for	
mset	
plate	
tape7	
dbplate	dir
tape6	
mplate	
oplate	
aplate	

all done

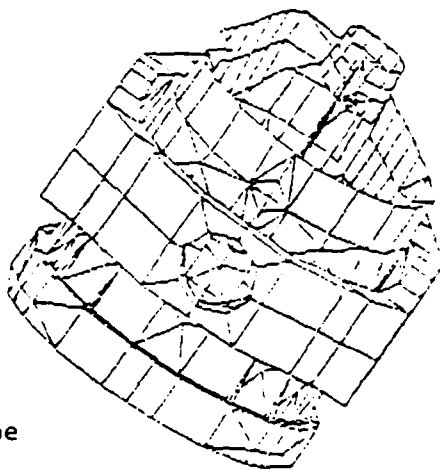
Demo Model



Wire Mesh Model



Hidden Line Model



Hidden Line Deformed Shape

Figure 3. PATRAN Renderings of Example Model.

ACRONYMS

CFS	Common File System - file storage for the CRAY (an IBM machine)
COSMOS	CRAY procedure language
CTSS	CRAY operating system
DMAP	Direct Matrix Abstraction Program - Program used to alter NASTRAN execution (usually for modified output)
GATEWAY	Program and network to transfer models and output from VAX to CRAY and CRAY to VAX
NASALTR	File containing DMAP statements for various outputs from NASTRAN
NASGO	Procedure program to set up and submit NASTRAN job
NASGO2	Procedure program to submit NASPAT job
NASJOB1	Input procedure modified by NASGO to execute NASTRAN
NASJOB2	Input procedure modified by NASGO2 to execute NASPAT
NASPAT	Translation program to convert NASTRAN results to PATRAN format
NASPRGEX	Program to add user parameters to NASTRAN input
NASTRAN	Multi-purpose finite element analysis program
MODAL.COM	Procedure program to convert models and results to appropriate format
MODAL.FOR	Program to convert PATRAN ASCII results file into PATRAN binary results file
PATRAN	Pre- and post-processor for NASTRAN

FILE TRANSFER AND PATRAN EXECUTION ON THE SILICON GRAPHICS

The files shown as output from NASTRAN on the CRAY (NASPTDAT) and the NASTRAN source file need to be transferred to the Silicon Graphics, from the VAX. The files must not be in STEXT or binary format. NASPTDAT is in STEXT format once it has been retrieved from mass storage to the VAX. NTEXT the file and call it some filename you can remember for transfer to the Silicon Graphics. If the source deck for NASTRAN is on the VAX in ASCII format (non-binary and non-STEXT) then it is also ready for transfer to the Silicon Graphics. If the file is only resident on the CRAY, then it must be put in STEXT format on the CRAY, stored in mass storage, retrieved on the VAX, and put in NTEXT format on the VAX prior to transfer to the Silicon Graphics. The following writeups are examples of how to retrieve files and transfer them to the Silicon Graphics. The discussion of how to run PATRAN on the Silicon Graphics is not covered here. Executing PATRAN is covered in the "PATRAN User's Guide".

To transfer files from the VAX to the Silicon Graphics:

While on the Silicon Graphics:

Type: connect

Log in on the VAX

Type: write sys\$output "~>:filename-on-iris"

The following two lines of commands will not be echoed to the screen!

Type: filename-on-VAX

Type: write sys\$output "~>"

Logoff the VAX.

Type: ~.

This will transfer the file from the VAX to the IRIS. Once there the file is ready for additional processing. This processing is performed with either NASPAT (for the source deck) or MODAL (for the results file).

To translate a source file to PATRAN format:

Type: NASPAT

Follow the prompts to translate a source deck into PATRAN format.

To translate a results file to PATRAN format:

Type: MODAL

Follow the prompts to translate the results file into the separate animation files in PATRAN format.

APPENDIX

Modal Strain Energy Tabulation (MSET)

1. Introduction

The modal strain energy method was developed for design analysis of viscoelastic damping treatments (Refs. A-1, A-2, A-3). Using this method, the designer seeks to maximize the modal strain energy in the viscoelastic treatment as a fraction of the total strain energy. While this information has been available from MSC/NASTRAN for many years in printed form, the form is not convenient. In addition to the total viscoelastic strain energy fraction, designers typically want to sum strain energies in several separate regions where damping may be applied. These calculations are typically required for several normal modes.

In response to this need, a program called MSET ("Modal Strain Energy Tabulation") was written (Ref. A-4). While the program is intended primarily for damping design, it is also useful in other normal modes analyses for evaluating mode shapes. MSET runs as a post processing step after a NASTRAN Solution 63 analysis. The user specifies groups of elements, such as viscoelastic areas, constraining layers, etc. Modal strain energies for all modes, or selected modes, are then broken down into the specified groups and displayed in tabular form. As an alternative, users may also display kinetic energy fractions organized by groups of node points to further aid in evaluating mode shapes. The code operates by reading user commands either interactively or in batch mode and producing a tabulated listing as a result. After the NASTRAN run has been completed, MSET may be run repeatedly with different user input.

MSET requires a DMAP alter, which is inserted into the NASTRAN Solution 63 normal modes run. This alter causes NASTRAN to write binary files containing modal strain energies and other information. MSET reads these files and accepts commands either interactively or from a batch mode command file, which cause it to output tabulated data on another file.

2. Executing MSET

MSET was originally written for VAX computers, with NASTRAN and MSET both running on the same machine. At the Air Force Weapons Laboratory (AFWL), NASTRAN runs on the Cray. It might have been possible to install MSET on the Cray for interactive execution on that machine. However, it was felt that interactive execution on a VAX would be more convenient, especially when graphic displays are produced. This choice also made it unnecessary to convert all of MSET from VAX to Cray Fortran. Furthermore, the focus of the present contract was on finding better ways for NASTRAN users to use the VAX and Cray computers cooperatively, and to transfer data between them.

MSET as originally written used binary files written by NASTRAN in its OUTPUT2 and OUTPUT4 formats. Running MSET entirely on the AR VAX would require transmission of the binary files written by NASTRAN from the Cray to the VAX. One aspect of this project was to investigate transmission of binary files between the Cray and the VAX, as documented in Reference A-5. Although the NOSTRADAMUS code (which is intended to transmit binary files) was obtained, this approach was not followed. Thus it was necessary to write a small Fortran code, using parts of MSET, and install it on the Cray. This code simply reads the binary files created by NASTRAN and writes the data to an ASCII file which is transferred to the VAX using the normal STEXT/mass-store/mass-get/NTEXT sequence discussed in Reference A-5. This post processing step is scheduled in the same batch

run with NASTRAN, provided the user requests MSET processing when setting up the run using NASGO.

After the run has been completed successfully, the user can return to the VAX and retrieve the ASCII file and run MSET either interactively or in batch mode.

3. Using MSET on the VAX

MSET runs on VAX computers and is normally invoked by the DCL command "MSET." This symbol must be defined either in a user's LOGIN.COM file or in the system-wide login file. MSET may be run either interactively or in batch mode. In interactive mode, commands are typed directly at the terminal. In batch mode, commands are inserted in an indirect command file which MSET reads. Most users prefer batch mode because they often want to repeat the same set of commands for different runs or make minor changes with a text editor.

For interactive mode, the DCL command is simply

```
$ MSET
```

Using batch mode, one writes commands into a file with the extension .IND and submits this file to MSET by typing

```
$ MSET filename
```

where `filename` is the command file name, without the .IND extension. An optional qualifier, `queue`, is the name of the desired batch queue (default SYSSBATCH).

```
$ MSET filename/QUE=queue
```

MSET operates by building tables of strain energies in which each row represents a set of elements and each column represents a particular normal mode. As an

option, MSET will sort elements in order of decreasing strain energy and print another table showing the first few elements in the sorted list (see SET DENSITY_NUMBER below). If kinetic energies are calculated, MSET will also sort grid points in order of decreasing kinetic energy and display the first few grids in this table (see SET KE). Tables are built internally and are then written to a file which can be printed or displayed.

MSET commands should be given in the following sequence:

1. The NASFILE command, to get MSET to read data from NASTRAN.
2. Optional commands to generate headers for the table: TITLE, SUBTITLE, VENT, VEMG, CLT, CLE, SET string.
3. The MODES command, to select a set of normal modes for which strain energies are to be calculated.
4. One or more SUM commands or a NASSET command to select a set of elements to use in calculating a single row of the table.
5. An optional LABEL command used to label the row being generated.
6. A CALCULATE command to cause MSET to carry out the calculation for a single row, using elements chosen by SUM or NASSET commands, and enter the row in the table. Any number of rows may be generated by entering successive sets of SUM or NASSET commands (and optional LABEL commands) followed by a CALCULATE command.
7. A STORE command to write the internally generated table to a file. After the STORE command, a new table may be generated by returning to step 1 or step 2.
8. An EXIT command must always be the final command.

At any point between the NASSET and EXIT commands, various SET and SHOW commands may be entered. Also, comment lines may appear at any point, beginning with an exclamation point (!).

As mentioned before, MSET may also be used to tabulate kinetic energies. In this case, the procedure is the same except that GRID commands are used instead of SUM commands to select sets of grid points instead of sets of elements, and the PUT_KE command is used instead of the CALCULATE command to store an individual line in the table.

Following are detailed descriptions of individual MSET commands:

NASFILE filename

The filename is the complete name (including extension) of the ASCII file that has been transferred from the Cray. MSET looks for filename.OUS and filename.MAS. The first file is required. It contains both LAMA and an ONRGY1 table and is written by NASTRAN DMAP alters in SOLution 63. A NASFILE command must appear prior to any other commands. If a filename.MAS is also found then the structure's mass and center of gravity will be calculated and displayed in the table. Filename.MAS must exist for kinetic energy calculation.

NASSET filename

filename is the name of a .SET file through which element numbers may be specified.

MODES 1,2,3,7,8

MODES 2 THRU 8 EXCEPT 5

MODES ALL

MODES selects modes for which strain energies will be calculated.

KEMODES 1,2,3,7,8

KEMODES 2 THRU 8 EXCEPT 5

KEMODES ALL

KEMODES selects modes for which kinetic energies will be calculated.

TITLE 'string'

string is a title for the table (up to 128 characters). "Percent Strain Energies" is the default title.

SUBTITLE 'string'

SUBTITLE 'string' n

'string' is a subtitle for the table (up to 128 characters). If a number n follows string, then string will be the n'th subtitle. Otherwise, successive subtitles will appear in the output table in the order in which the SUBTITLE commands are given.

LABEL 'string'

LABEL 'string' n

string is a label for a table row (up to 32 characters). If a number n follows string, then string will label the row corresponding to "n." Otherwise, string will label the row following the last calculated row.

SUM 1000,1010,1020

SUM 10 THRU 500 EXCEPT 100 THRU 200

SUM ALL EXCEPT 50 THRU 1000,5000,5050

SUM HEXA

SUM SET n

SUM HEXA EXCEPT 1050,1060,1070

SUM QUAD4,BAR

SUM sets flags in the element table to include or exclude particular elements in a calculation. The SUM SET command above refers to set n within a NASSET file that has previously been opened with the NASSET command. It will flag all elements within set n for inclusion within the next calculation. All other numbers used in the SUM command are element numbers.

SHOW TABLE
SHOW MODE
SHOW TITLE
SHOW LABEL
SHOW SUM
SHOW GRIDS
SHOW KE
SHOW LAST_COMMAND
SHOW LAST_MESSAGE
SHOW DATE
SHOW TIME

SHOW lets the user examine the table and various states from within the program. SHOW TABLE shows the table exactly as it would appear if sent to a printer. SHOW MODE shows the modes within the NASTRAN file, as well as which modes will be included the table and calculations. SHOW SUM shows which elements have been detected in the NASTRAN file, as well as each element's status for inclusion in the next calculation. SHOW GRIDS shows the grids found within the NASTRAN file, as well as which grids will be included in the table with the PUT_KE command. SHOW KE shows the percent kinetic energies for the grids chosen with the GRIDS command.

CALCULATE

CALCULATE sums the percent strain energies for all flagged elements for each flagged mode and then clears all the element flags prior to the next calculation.

GRIDS 1000,1010,1020

GRIDS 100 THRU 999

GRIDS ALL EXCEPT 2000 THRU 50000

GRIDS sets flags in the grid table to include, or not include, grids in the table when the PUT_KE command is given. Note, the setting of the grid flags will not affect the calculation of percent kinetic energies, i.e., percent kinetic energies are calculated using all grid points (but no scalar points). GRIDS has no effect on the output of the maximum percent kinetic energies as designated by the SET KE command below.

PUT_KE

PUT_KE inserts into the table a listing of the percent kinetic energies of all grids set by the GRIDS command for all modes set by KEMODES. The kinetic energies at each of the six degrees of freedom for each gridpoint are displayed.

STORE filename

filename is the name of a file where the percent strain energy table is stored.

DELETE n

n is the row number in the percent strain energy table that is permanently removed from the table.

SET PROMPT 'string'

SET DENSITY_NUMBER n

SET KE m

string is a character string to be used as the program prompt (up to eight characters). The number n controls the second part of the table, in which strain energies are sorted and the first n elements are printed. Similarly, m controls the third part,

in which kinetic energies (if any) are sorted, and the *m* points having maximum values are printed. The SET KE command is independent of the GRIDS and PUT KE commands, i.e., a particular grid point may be printed more than once per mode in the table if it has the highest absolute kinetic energies in more than one degree of freedom. (*m*=0, *n*=0, and *string*='*' are the defaults.)

```
VENT [n] = number
VEMG [n] = number
CLT [n] = number
CLE [n] = number
SET string = number
```

These commands will report the value of the indicated viscoelastic property at the top of the strain energy table under the "Mass =" line. *string* is any property variable name. *number* is the value to be printed, and it must appear in standard "F," "E," or "I" format. *n* is an optional subscript value between 1 and 5 used to display values for more than one of a particular property.

EXIT

EXIT exits the program.

@filename

filename is the name of an indirect command file that may contain MSET commands. The file name must have the extension ".IND." Any line within *filename*.IND that starts with an exclamation point (!) is considered a comment line and is ignored.

4. Special Instructions for MSET

The following special instructions must be observed in using MSET on the AFWL Cray and the AR VAX computers:

1. The MSET option (option 12) must be chosen when running NASGO. Reference A-5 explains NASGO.
2. SOLution 63 must be selected.
3. The MSET alter for version 63 must be included. The alter is available as part of the alter library used by NASGO, and instructions on the use of alters may be found in Reference A-5.
4. The case control deck must include the line

ESE=ALL

The alter suppresses printout of element strain energies by changing the default value of PARAM,TINY to 99 percent. If element strain energy printout is desired, PARAM,TINY should be set back to a small value such as 10^{-3} . See section 3.1 of Reference A-6 for more about PARAM,TINY.

5. After the NASTRAN run has been concluded, the COSMOS control program invoked by NASGO will have stored the MSET data file in mass storage under the name Mjob, where "job" is the name of the job that was specified to NASGO. That is, if the job name was PLATE, then MPLATE will be stored. This file must be fetched back to the AR VAX using MASS and converted using NTEXT.

5. Typical Output

Figure 1 shows PLATE.IND, a typical input file for MSET. To run this, one would type

```
$ MSET PLATE
```

or

```
$ MSET
```

```
@PLATE
```

```
EXIT
```

Figure 2 shows strain energy output as requested by the SUM commands. Figure 3 shows sorted strain energies, one row for each mode. Figure 4 shows kinetic energies sorted by individual degrees of freedom and also by grid point.

```

INASTRAN plate model
!Open and read the data files
naaf plate
title 'Plate percent strain energies'
subt 'This plate is a simple test model'
subt 'Face sheets are aluminum, VEM is ISO112'
!Display the damping model properties
vent = 0.005
vemg = 150
c1t = 0.035
c1e = 1.E+07
!Calculate strain energies for all modes
mode all
!Display the top ten strain energy densities in the table
set dens 10
!Sum percent strain energies based on element #'s
sum all except 1 thru 4
lab 'all except 1 thru 4'
calc
sum 1,3,5,7,9,11,13,15
lab 'odd elements'
calc
!Sum percent strain energies based on element type
sum quad4
lab 'all quad4 elements'
calc
!### Kinetic energy commands ###
!Only calculate percent kinetic energies for modes 1 and 2
kemode 1,2
!Display the top ten percent kinetic energies in the table
set ke 10
!Display all 6 dof percent kinetic energies for gridpoints 1,2,8,9,10
grids 1,2,8,9,10
put_ke
!Store the table in PLATE.ESE
store plate.eae
!Exit from MSET
ex

```

Figure 1. Typical MSET input

Plate percent strain energies
 This plate is a simple test model
 Face sheets are aluminum, VEM is ISD112

Mass = 7.290E+00
 VEM(1) = 5.000E-03
 VEM(2) = 1.500E+02
 CLT(1) = 3.500E-02
 CLE(1) = 1.000E+07

Frequency (Hz)	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7	Mode 8	Mode 9	Mode 10
all except 1 thru 4	7.750E+01	1.939E+02	4.350E+02	5.543E+02	6.724E+02	1.011E+03	1.235E+03	1.272E+03	1.564E+03	1.711E+03
odd elements	7.573E+01	7.185E+01	7.302E+01	8.372E+01	6.978E+01	6.678E+01	7.325E+01	7.587E+01	8.563E+01	8.888E+01
all quad elements	7.278E+01	5.026E+01	6.975E+01	4.502E+01	4.529E+01	3.639E+01	3.521E+01	4.786E+01	4.119E+01	3.025E+01
	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02

Frequency (Hz)	Mode 11
all except 1 thru 4	1.858E+03
odd elements	6.606E+01
all quad elements	3.212E+01

X c.g. = 5.137E+00 Y c.g. = 0.000E+00 Z c.g. = 0.000E+00
 Table created: 13-SEP-88 00:35:21 from [EXAMPLES]PLATE.DDS:2

Figure 2. Typical MSET strain energy table

Strain Energy Densities

Mode 1 ELEMENT ELEMENT S.E.D.	8 QUAD4 1.341E+04	5 QUAD4 1.341E+04	1 QUAD4 1.166E+04	13 QUAD4 1.166E+04	2 QUAD4 5.292E+03	14 QUAD4 5.292E+03	6 QUAD4 4.697E+03	10 QUAD4 4.697E+03	3 QUAD4 1.258E+03	15 QUAD4 1.258E+03
Mode 2 ELEMENT ELEMENT S.E.D.	13 QUAD4 5.541E+04	1 QUAD4 5.541E+04	2 QUAD4 3.339E+04	14 QUAD4 3.339E+04	6 QUAD4 3.258E+04	10 QUAD4 3.258E+04	7 QUAD4 3.018E+04	11 QUAD4 3.018E+04	3 QUAD4 2.948E+04	15 QUAD4 2.948E+04
Mode 3 ELEMENT ELEMENT S.E.D.	15 QUAD4 2.321E+05	3 QUAD4 2.321E+05	7 QUAD4 1.991E+05	11 QUAD4 1.991E+05	1 QUAD4 1.928E+05	13 QUAD4 1.928E+05	8 QUAD4 1.594E+05	12 QUAD4 1.594E+05	2 QUAD4 1.311E+05	14 QUAD4 1.311E+05
Mode 4 ELEMENT ELEMENT S.E.D.	12 QUAD4 4.905E+05	8 QUAD4 4.905E+05	7 QUAD4 4.477E+05	11 QUAD4 4.477E+05	6 QUAD4 2.287E+05	10 QUAD4 2.287E+05	2 QUAD4 2.119E+05	14 QUAD4 2.119E+05	3 QUAD4 2.042E+05	15 QUAD4 2.042E+05
Mode 5 ELEMENT ELEMENT S.E.D.	12 QUAD4 6.400E+05	8 QUAD4 6.400E+05	3 QUAD4 5.295E+05	15 QUAD4 5.295E+05	4 QUAD4 4.152E+05	16 QUAD4 4.152E+05	2 QUAD4 4.013E+05	14 QUAD4 4.013E+05	1 QUAD4 3.609E+05	13 QUAD4 3.609E+05
Mode 6 ELEMENT ELEMENT S.E.D.	18 QUAD4 1.962E+06	4 QUAD4 1.962E+06	8 QUAD4 9.559E+05	12 QUAD4 9.559E+05	3 QUAD4 8.537E+05	15 QUAD4 8.537E+05	2 QUAD4 7.397E+05	14 QUAD4 7.397E+05	1 QUAD4 7.380E+05	13 QUAD4 7.380E+05
Mode 7 ELEMENT ELEMENT S.E.D.	18 QUAD4 2.314E+06	4 QUAD4 2.314E+06	8 QUAD4 2.205E+06	12 QUAD4 2.205E+06	3 QUAD4 1.580E+06	15 QUAD4 1.580E+06	7 QUAD4 1.307E+06	11 QUAD4 1.307E+06	2 QUAD4 1.029E+06	14 QUAD4 1.029E+06
Mode 8 ELEMENT ELEMENT S.E.D.	12 QUAD4 1.516E+06	8 QUAD4 1.516E+06	4 QUAD4 1.456E+06	16 QUAD4 1.456E+06	6 QUAD4 1.354E+06	10 QUAD4 1.354E+06	5 QUAD4 1.293E+06	9 QUAD4 1.293E+06	3 QUAD4 1.259E+06	15 QUAD4 1.259E+06
Mode 9 ELEMENT ELEMENT S.E.D.	14 QUAD4 3.122E+06	2 QUAD4 3.122E+06	4 QUAD4 2.988E+06	16 QUAD4 2.988E+06	3 QUAD4 2.750E+06	15 QUAD4 2.750E+06	8 QUAD4 2.331E+06	12 QUAD4 2.331E+06	1 QUAD4 1.757E+06	13 QUAD4 1.757E+06
Mode 10 ELEMENT ELEMENT S.E.D.	18 QUAD4 5.883E+06	4 QUAD4 5.883E+06	2 QUAD4 2.861E+06	14 QUAD4 2.861E+06	8 QUAD4 2.185E+06	12 QUAD4 2.185E+06	6 QUAD4 2.165E+06	10 QUAD4 2.165E+06	3 QUAD4 2.098E+06	15 QUAD4 2.098E+06
Mode 11 ELEMENT ELEMENT S.E.D.	16 QUAD4 5.867E+06	4 QUAD4 5.867E+06	2 QUAD4 3.653E+06	14 QUAD4 3.653E+06	3 QUAD4 3.591E+06	15 QUAD4 3.591E+06	12 QUAD4 3.233E+06	8 QUAD4 3.233E+06	6 QUAD4 2.019E+06	10 QUAD4 2.019E+06

Figure 3. Sorted strain energy output

Maximum Percent Kinetic Energies

Mode 1						
GRID #	DOF	%KE				
9	3	0.21066				
8	3	0.10893				
10	3	0.10893				
22	3	0.09407				
19	3	0.09199				
25	3	0.09199				
2	3	0.05239				
3	3	0.05239				
7	3	0.04326				
11	3	0.04326				
Mode 2						
GRID #	DOF	%KE				
11	3	0.14479				
7	3	0.14479				
2	3	0.12279				
3	3	0.12279				
19	3	0.07454				
25	3	0.07454				
8	3	0.06603				
10	3	0.06603				
6	3	0.05479				
12	3	0.05479				
Mode 1						
GRID #	T1	T2	T3	R1	R2	R3
1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.05239	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.10893	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.21066	0.00000	0.00010	0.00000
10	0.00000	0.00000	0.10893	0.00000	0.00000	0.00000
Mode 2						
GRID #	T1	T2	T3	R1	R2	R3
1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.12279	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.06603	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00050	0.00000	0.00000
10	0.00000	0.00000	0.06603	0.00000	0.00000	0.00000

Figure 4. Sorted kinetic energy output

6. References

- A-1 Johnson, C.D. and Kienholz, D.A., "Finite Element Prediction of Damping in Structures with Constrained Viscoelastic Layers," *AIAA J.*, Vol. 20, No. 9, September 1982.
- A-2 Johnson, C.D., Kienholz, D.A., and Parekh, J.C., "Design Methods for Viscoelastically Damped Plates," *Proc. 24th Structures, Structural Dynamics and Materials Conf.*, Lake Tahoe, NV, May 1983.
- A-3 Johnson, C.D., "Design and Analysis of Damped Structures using Finite Element Techniques," ASME Paper No. 85-DET-131, Conference on Mechanical Vibration and Noise, Cincinnati, OH, September 1985.
- A-4 Fowler, B.L., "MSET User's Manual," CSA Engineering Report No. 87-04-04, April 1987.
- A-5 Applied Technology Associates Memo SBIR-0022, "VAX to Cray NASTRAN User Interface," July 1987.
- A-6 MacNeal-Schwendler Corp., *MSC/NASTRAN User's Manual*