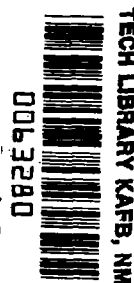


NASA  
RP  
1019  
c.1

NASA Reference Publication 1019

LOAN COPY: RETURN TO  
AFWL TECHNICAL LIBRARY  
KIRTLAND AFB, NM



# Guide to a Condensed Form of NASTRAN®

James L. Rogers, Jr.

SEPTEMBER 1978

**NASA**

TECH LIBRARY KAFB, NM



0063280

NASA Reference Publication 1019

# Guide to a Condensed Form of NASTRAN<sup>®</sup>

James L. Rogers, Jr.  
*Langley Research Center  
Hampton, Virginia*

**NASA**

National Aeronautics  
and Space Administration

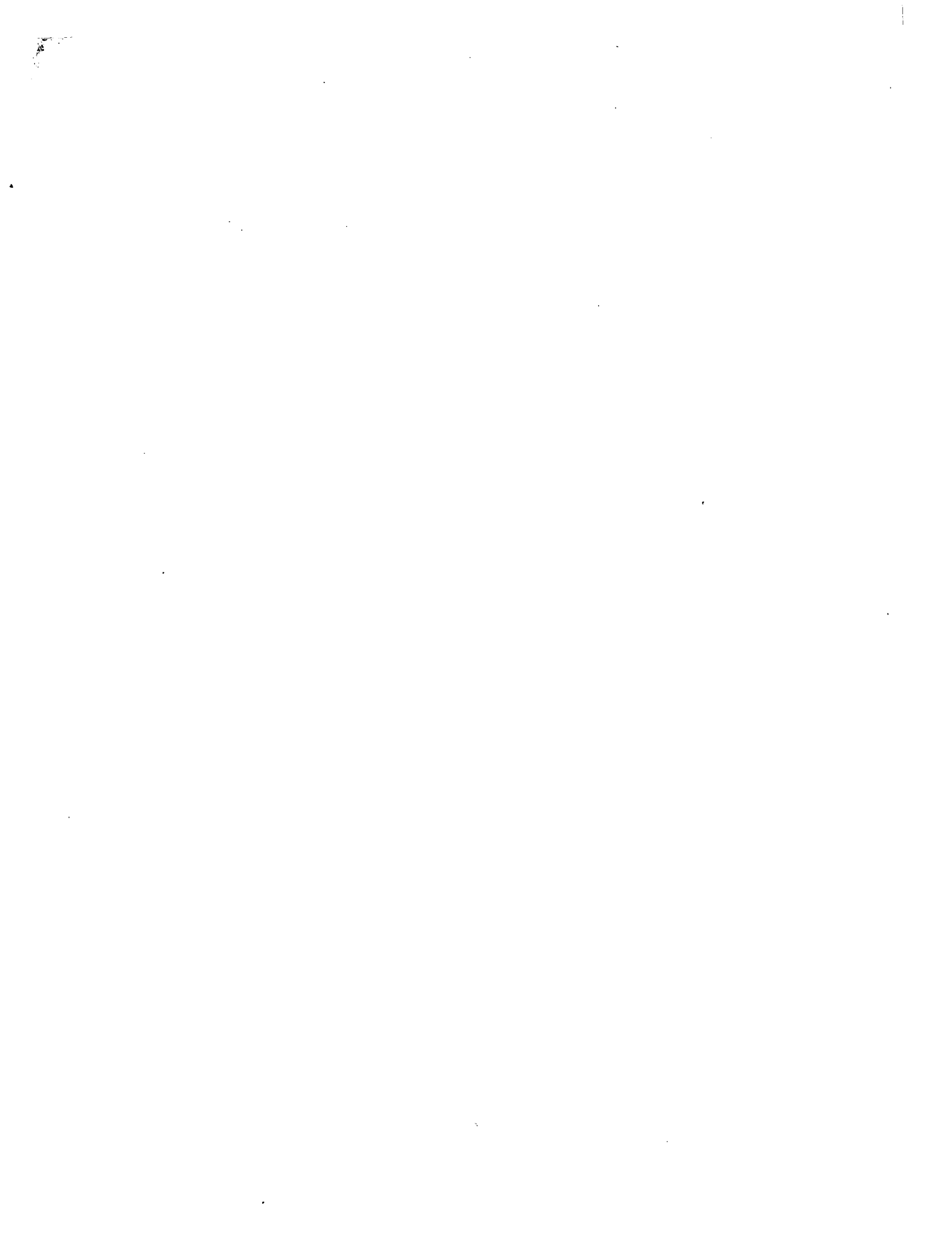
**Scientific and Technical  
Information Office**

1978



## PREFACE

A condensed form of NASTRAN® Level 16 is described. Included are descriptions of the input cards, the programming language of the direct matrix abstraction program, the plotting, the problem definition, the modules' diagnostic messages, and sample problems that relate to the capabilities retained in the condensed form. This guide can serve as a handbook for instructional courses in the use of NASTRAN or for users who only need the capability provided by the condensed form.



# CONTENTS

PREFACE . . . . .	iii
INTRODUCTION . . . . .	1
1. STRUCTURAL MODELING . . . . .	2
1.1 Problem Definition . . . . .	2
1.2 Grid-Point Definition . . . . .	3
1.3 Grid-Point Sequencing . . . . .	5
1.4 Stiffness Element Definitions . . . . .	8
1.4.1 Bar Elements . . . . .	9
1.4.2 Shear Panel Elements . . . . .	11
1.4.3 Plate Elements . . . . .	13
1.5 Mass Element Definitions . . . . .	14
1.5.1 Lumped Mass . . . . .	14
1.5.2 Coupled Mass . . . . .	16
1.5.3 Mass Input . . . . .	16
1.5.4 Symbols Used in Figures 4, 5, and 7 . . . . .	17
1.6 Constraints and Partitioning . . . . .	18
1.6.1 Single-Point Constraints . . . . .	18
1.6.2 Free-Body Supports . . . . .	19
1.6.3 Partitioning . . . . .	20
1.7 Static Loads . . . . .	20
1.8 Analyses Descriptions . . . . .	21
1.8.1 Linear Static Analysis . . . . .	21
1.8.2 Vibration Analysis . . . . .	22
1.8.3 Buckling Analysis . . . . .	22
1.8.4 Direct Matrix Abstraction Programs . . . . .	22
2. NASTRAN DATA DECK . . . . .	23
2.1 General Description of NASTRAN Data Deck . . . . .	23
2.2 Bulk Data Deck . . . . .	24
2.3 Case Control Deck . . . . .	65
2.4 Executive Control Deck . . . . .	87
2.5 NASTRAN Card . . . . .	87

3. PLOTTING . . . . .	88
3.1 General Description of Plotting Capability . . . . .	88
3.2 Structure Plotting . . . . .	91
3.2.1 Rules for Free-Field Card Specification . . . . .	92
3.2.2 Plot Request Packet Card Format . . . . .	92
3.2.3 Plot Titles . . . . .	93
3.2.4 Plot Request Packet Card Descriptions . . . . .	93
3.2.5 SET Definition Cards . . . . .	93
3.2.6 Cards Defining Parameters . . . . .	95
3.2.7 Summary of Structure Plot Request Packet Cards . . . . .	107
3.3 NASTRAN General Purpose Plotter . . . . .	109
4. DIRECT MATRIX ABSTRACTION PROGRAMMING . . . . .	113
4.1 General DMAP Rules . . . . .	113
4.2 DMAP Rules . . . . .	113
4.3 DMAP Rules for Functional Module Instructions . . . . .	114
4.4 Functional Module Input Data Blocks . . . . .	115
4.5 Functional Module Output Data Blocks . . . . .	115
4.6 Functional Module Parameters . . . . .	115
4.7 DMAP Compiler Options - XDMAP Instruction . . . . .	118
4.8 Extended Error Handling Facility . . . . .	119
4.9 DMAP Rules for Executive Operation Instructions . . . . .	119
4.10 Techniques and Examples of Executive Module Usage . . . . .	120
4.11 REPT and FILE Instructions . . . . .	120
4.12 EQUIV Instruction . . . . .	122
4.13 PURGE Instruction . . . . .	124
5. DIRECT MATRIX ABSTRACTION MODULES . . . . .	128
5.1 Index of DMAP Modules Descriptions . . . . .	128
5.2 Executive Operation Modules . . . . .	129
5.3 Structurally Oriented Modules . . . . .	142

5.4	Matrix Operation Modules . . . . .	184
5.5	Utility Modules . . . . .	202
6.	ERROR MESSAGES . . . . .	219
6.1	NASTRAN System and User Messages . . . . .	219
6.2	Preface User Messages . . . . .	220
6.3	Functional Module Messages . . . . .	227
6.4	Analysis Error Messages . . . . .	240
6.4.1	Error Messages for Static Analysis . . . . .	240
6.4.2	Error Messages for Vibration Analysis . . . . .	241
6.4.3	Error Messages for Buckling Analysis . . . . .	241
	APPENDIX - SAMPLE PROBLEMS . . . . .	243
	REFERENCES . . . . .	290



## INTRODUCTION

NASTRAN<sup>1</sup> (NASA Structural Analysis) is a finite-element computer program for structural analysis and is intended for general use. As such, it must answer to a wide spectrum of requirements and must permit future modifications and continued expansion to new problem areas. But because of its size, cost, generality, and voluminous documentation, NASTRAN has not gained popularity among universities and small consulting firms. For the most part, these organizations neither need all the capability NASTRAN provides nor can they afford the cost of more recent levels. To provide a form of NASTRAN compatible with their needs, a condensed NASTRAN was created. This condensed form of NASTRAN is simply a limited capability form of Level 16.

Capabilities for a condensed form of NASTRAN were selected after discussions with engineers, programmers, and university professors who are familiar with NASTRAN. The three types of analysis that appeared to be most widely favored were

Linear static analysis

Vibration analysis

Buckling analysis

The user-oriented language used to perform these analyses in the condensed form of NASTRAN is called DMAP (direct matrix abstraction program). DMAP instructions consist of two types: functional modules and executive operations. Functional modules are arbitrarily classified as structural modules, matrix operation modules, and utility modules. All remaining modules are classified as executive operations. With DMAP, the user can go beyond the scope of the three previously mentioned structural analyses and can also solve a broad range of nonstructural matrix problems, all within the framework of NASTRAN. In order to use this condensed form of NASTRAN, the user is expected to be familiar with the rudiments of finite-element structural analysis and to be aware of basic modeling concepts. A detailed knowledge of computer programming is not required although some training will help.

The user must first define the structural model by using finite elements. Information on problem definition is given in section 1.1. The number of cards (executive, case control, and bulk data) that the user can input to the condensed form of NASTRAN in order to describe the model has been limited. Descriptions of these cards can be found in section 2. Only four finite elements are available for structural modeling; they include BAR (single beam), SHEAR (shear panel), TRIA2 (triangular membrane and bending), and TRIMEM (triangular membrane). In addition, the CONM2 (concentrated mass) is also included. Only one method of eigenvalue extraction (i.e., inverse power), can be selected for vibration and buckling analysis. The user can obtain a graphi-

---

<sup>1</sup>NASTRAN: Registered trademark of the National Aeronautics and Space Administration.

cal display of the structural model by using the NASTRAN plot package. The Stromberg-Carlson, CalComp, and NASTRAN general purpose plotters are permitted. Section 3 contains complete details of the plotting capability. The user has 58 modules with which to do DMAP programming. Rules for using these modules in DMAP programming are found in section 4, and the modules themselves are described in section 5. Many diagnostic messages are provided in section 6 to help the user. Finally, four sample problems are given in the appendix.

Four technical manuals document the NASTRAN Level 16 program. The NASTRAN Theoretical Manual (ref. 1) presents a topic-by-topic discourse on the theory, assumptions, and methods of analysis for those interested in the background of NASTRAN. The NASTRAN User's Manual (ref. 2) is devoted to describing those items related to the use of NASTRAN that are independent of the computing system being used. The third document is The NASTRAN Programmer's Manual (ref. 3) which describes the details of the program, its organization, its file structure, and its implementation on each of three computer systems (i.e., IBM 360-370, UNIVAC 1100 Series, and Control Data 6000 Series) for which NASA maintains NASTRAN. The NASTRAN Demonstration Problem Manual (ref. 4) illustrates the types of problems that can be solved with NASTRAN and shows that the results are valid.

The present publication serves only as a guide for using the condensed form of NASTRAN. Additional information required with respect to the programming, user, or theoretical aspects of NASTRAN is available in the previously mentioned manuals.

Any reference made to NASTRAN within the text refers to the condensed form of NASTRAN unless specifically stated otherwise.

Identification of commercial products in this report is used to adequately describe the model. The identification of these commercial products does not constitute official endorsement, expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

## 1. STRUCTURAL MODELING

### 1.1 Problem Definition

The grid-point definition forms the basic framework for the structural model. All other parts of the structural model are referenced either directly or indirectly to the grid points. A grid point is a point in three-dimensional space at which three components of translation and three components of rotation are defined. The coordinates of each grid point are specified by the user.

The structural element is a convenient means for specifying many of the properties of the structure, including material properties, mass distribution, and some types of applied loads. In static analysis by the displacement method, stiffness properties are input exclusively by means of structural elements. Mass properties used in the generation of inertia loads are input either as properties of structural elements or as properties of grid points. In dynamic analysis, mass and stiffness properties may be input either as the properties

of structural elements or as the properties of grid points (direct input matrices).

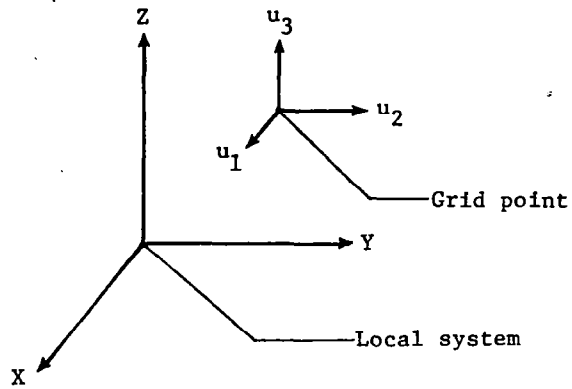
Structural elements are defined on connection cards by referencing grid points. Sometimes when NASTRAN is used, all the information required to generate the structural matrices for the element is given on the connection card. Most of the time, the connection card refers to a property card on which the cross-sectional properties of the element are given. The property card, in turn, refers to a material card which gives the material properties.

Single-point constraints are used to limit selected degrees of freedom to either zero or a prescribed value. Omitted points are used as a tool in matrix partitioning and to reduce the number of independent degrees of freedom in the matrix used for analysis. Free-body supports are used to remove stress-free motions in static analysis and to evaluate the free-body inertia properties of the structural model. Static loads are applied to the structural model by concentrated loads at grid points.

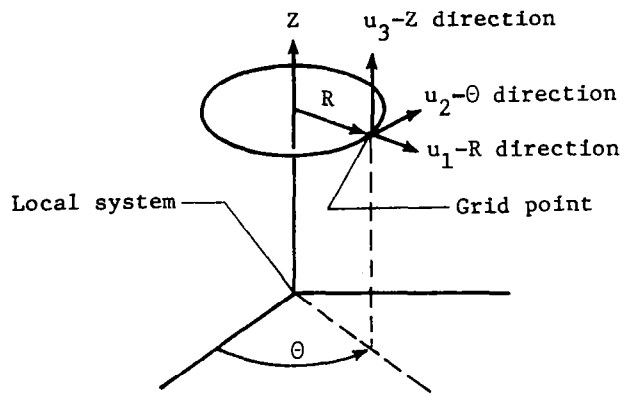
## 1.2 Grid-Point Definition

The description of a structural model is input to NASTRAN on data cards. Each data card contains a name or mnemonic which indicates the type of information on the card. Grid points are defined on GRID data cards (GRID is the mnemonic) by specifying coordinates in either the basic or a local coordinate system. The implicitly defined basic coordinate system is rectangular. Local coordinate systems may be rectangular, cylindrical, or spherical. Each local system must be related either directly or indirectly to the basic coordinate system. The CORD1C, CORD1R, and CORD1S cards are used to define cylindrical, rectangular, and spherical local coordinate systems, respectively, in terms of three grid points which have been previously defined.

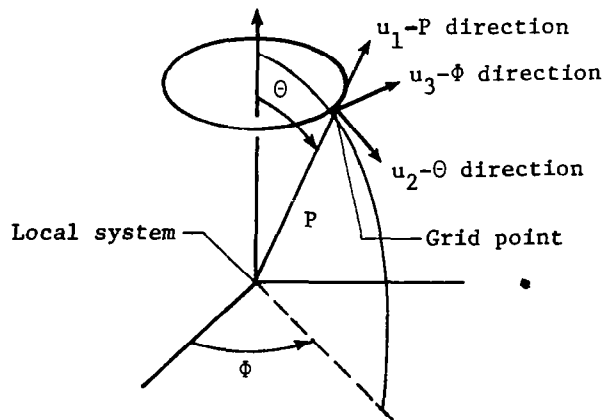
Six rectangular displacement components (three translations and three rotations) are defined at each grid point. The local coordinate system used to define the displacements may be different from the local coordinate system used to locate the grid point. Both the location coordinate system and the displacement coordinate system are specified on the GRID card for each geometric grid point. The orientation of displacement components depends on the type of local coordinate system used to define the displacement components. If the defining local system is rectangular, the displacement system is parallel to the local system and is independent of the grid-point location, as indicated in figure 1(a). If the local system is cylindrical, the displacement components are in the radial, tangential, and axial directions, as indicated in figure 1(b). If the local system is spherical, the displacement components are in the radial, meridional, and azimuthal directions, as indicated in figure 1(c). Each geometric grid point may have a unique displacement coordinate system associated with it. The collection of all displacement coordinate systems is known as the global coordinate system. All matrices are formed and all displacements are output in the global coordinate system. The symbols T1, T2, and T3 on the printed output indicate translations for each grid point in the 1, 2, and 3 directions, respectively. The symbols R1, R2, and R3 indicate rotations about the three axes.



(a) Rectangular.



(b) Cylindrical.



(c) Spherical.

Figure 1.- Displacement coordinate systems.

Provision is also made on the GRID card to apply single-point constraints to any displacement component. Any constraints specified on the GRID card are automatically used for all solutions. Constraints specified on the GRID card are usually restricted to those degrees of freedom that will not be elastically constrained and hence must be removed from the model in order to avoid singularities in the stiffness matrix.

### 1.3 Grid-Point Sequencing

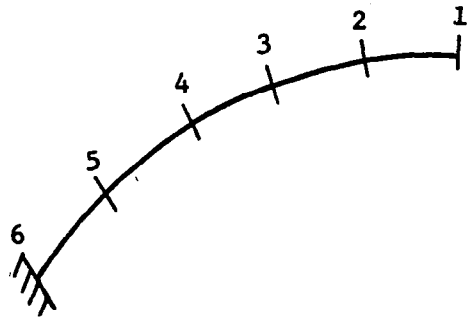
The best decomposition times and equation solution times are obtained if the grid points can be sequenced in such a manner as to create matrices having small numbers of active columns. The decomposition time is proportional to the sum of the squares of the number of active columns in each row of the triangular factor. The equation solution time (forward and backward substitution) is proportional to the number of nonzero terms in the triangular factor.

In selecting the grid-point sequencing, it is not important to find the best sequence; rather, it is usually satisfactory to find a good sequence and to avoid bad sequences that create unreasonably large numbers of active columns. For many problems, a sequence which results in a band matrix is a reasonably good choice but not necessarily the best. Also, sequences which result in small numbers of columns with nonzero terms are usually good but not necessarily the best. A sequence with a larger number of nonzero columns frequently has a smaller number of nonzero operations in the decomposition when significant passive regions exist within the active columns.

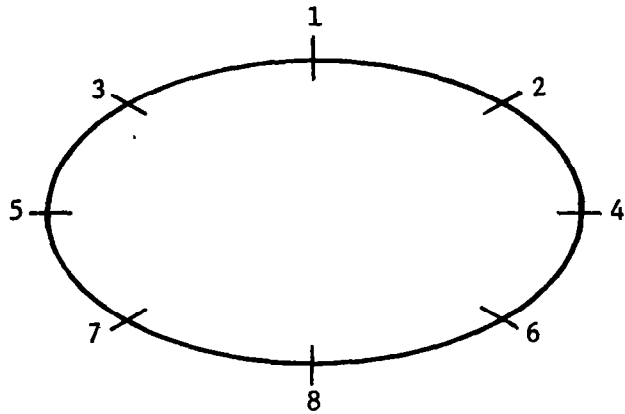
Examples of proper grid-point sequencing for one-dimensional systems are shown in figure 2. For open loops, a consecutive numbering system should be used, as shown in figure 2(a). This sequencing results in a narrow band matrix with no new nonzero terms created during the triangular decomposition. Generally, there is an improvement in the accumulated round-off error if the grid points are sequenced from the flexible end to the stiff end.

For closed loops, the grid points may be sequenced either as shown in figure 2(b) or figure 2(c). If the sequencing is as shown in figure 2(b), the semiband is twice that of the model shown in figure 2(a). The matrix initially contains a number of zeros within the band which become nonzero as the decomposition proceeds. If the sequencing is as shown in figure 2(c), the band portion of the matrix is the same as that for figure 2(a). However, the connection between grid points 1 and 8 creates a number of active columns on the right-hand side of the matrix. Solution times are the same for the sequence shown in figure 2(b) or 2(c) because the number of active columns in each sequence is the same.

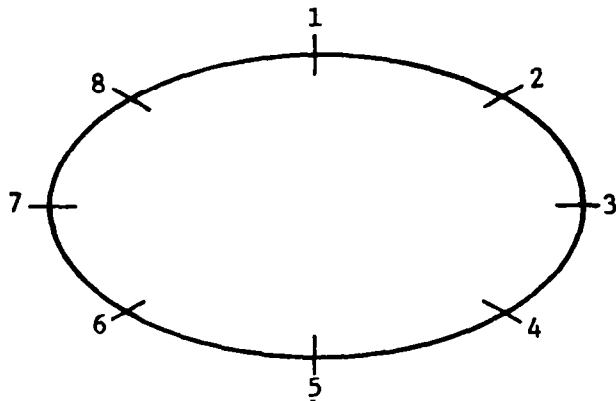
Examples of grid-point sequencing for surfaces are shown in figure 3. For plain or curved surfaces with a pattern of grid points that tends to be rectangular, the sequencing shown in figure 3(a) results in a band matrix that has good solution times. The semiband is proportional to the number of grid points along the short direction of the pattern. If the pattern of grid points shown in figure 3(a) is made into a closed surface by connecting grid points 1 and 17, 2 and 18, and so forth, a number of active columns equal to the semiband are



(a) Consecutive numbering system for open loops.

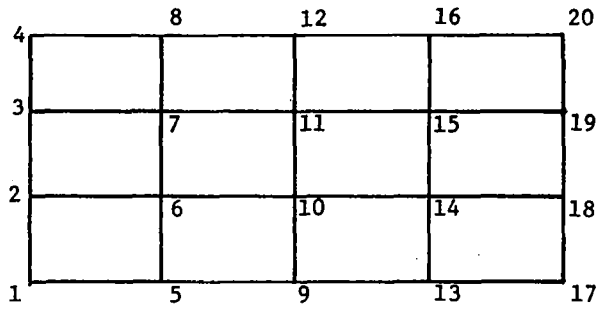


(b) Sequencing of grid points for a closed loop (method 1).

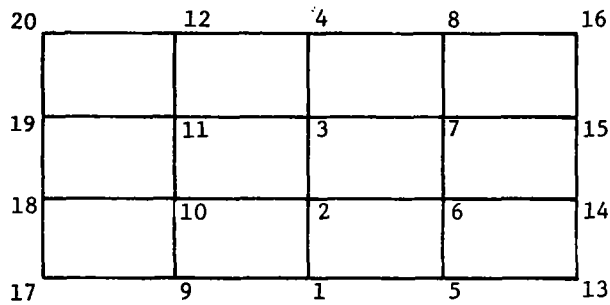


(c) Sequencing of grid points for a closed loop (method 2).

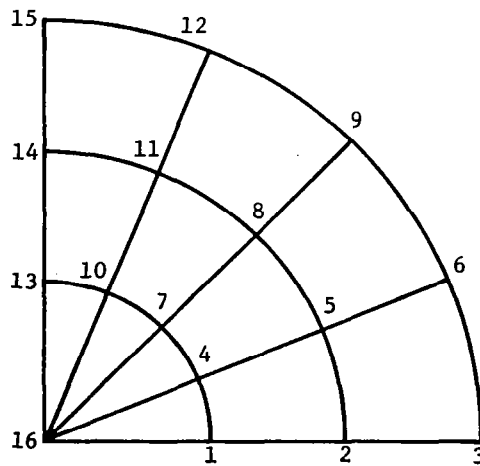
Figure 2.- Grid-point sequencing for one-dimensional systems.



(a) Grid-point sequencing for a rectangular surface (method 1).



(b) Grid-point sequencing for a rectangular surface (method 2).



(c) Grid-point sequencing for a radial pattern.

Figure 3.- Grid-point sequencing for surfaces.

created. If the number of grid points in the circumferential direction is greater than twice the number in the axial direction, the sequencing indicated in figure 3(a) is a good one. However, if the number of grid points in the circumferential direction is less than twice the number in the axial direction, the use of consecutive numbering in the circumferential direction is more efficient. An alternate sequencing for a closed loop is shown in figure 3(b), where the semiband is proportional to twice the number of grid points in a row. For cylindrical or similar closed surfaces, the sequencing shown in figure 3(b) has no advantage over that shown in figure 3(a) since the total number of active columns is the same for either sequencing pattern.

With the exception of the central point, sequencing considerations for the radial pattern shown in figure 3(c) are similar to those for the rectangular patterns shown in figures 3(a) and 3(b). The central point must be sequenced last to limit the number of active columns associated with this point to the number of degrees of freedom at the central point. If the central point is sequenced first, the number of active columns associated with the central point is proportional to the number of radial lines. If there are more grid points on a radial line than on a circumferential line, the consecutive numbering should extend in the circumferential direction, beginning with the outermost circumferential ring. In this case, the semiband is proportional to the number of grid points on a circumferential line and there will be no active columns on the right-hand side of the matrix. If the grid points form a full circular pattern, the closure creates a number of active columns proportional to the number of grid points on a radial line, provided that the grid points are numbered as shown in figure 3(c). Proper sequencing for a full circular pattern is similar to that discussed for the rectangular arrays shown in figures 3(a) and 3(b) for closed surfaces.

Sequencing problems for actual structural models can frequently be handled by considering that the model consists of several substructures. Each substructure is first numbered in the most efficient manner. The substructures are then connected so as to create the minimum number of active columns. The grid points at the interface between two substructures are usually given numbers near the end of the sequence for the first substructure and as near the beginning of the sequence for the second substructure as is convenient.

#### 1.4 Stiffness Element Definitions

Stiffness elements are defined on connection cards that identify the grid points to which the element is connected. The mnemonics for all such cards have a prefix C, followed by an indication of the type of element, such as CBAR. The order of the grid-point identification defines the positive direction of the axis of a one-dimensional element and the positive surface of a plate element. The connection cards include additional orientation information when required. Each connection card references a property definition card. If many elements have the same properties, this system of referencing eliminates a large number of duplicate entries.

The property definition cards define geometric properties such as thicknesses, cross-sectional areas, and moments of inertia. The mnemonics for all



such cards have a prefix P, followed by all the characters used on the associated connection card, such as PBAR. Other included items are the nonstructural mass and the location of points where stresses will be calculated. Each property definition card references a material property card.

For plate elements, a different property card is provided for each type of element such as membrane plates. Thus, each property card contains only the information required for a single type of plate element, and usually a single card has sufficient space for all property information. To maintain uniformity in the relationship between connection cards and property cards, a number of connection card types contain the same information, such as the connection cards for the various types of triangular elements. Property cards for triangular elements of the same type contain the same information.

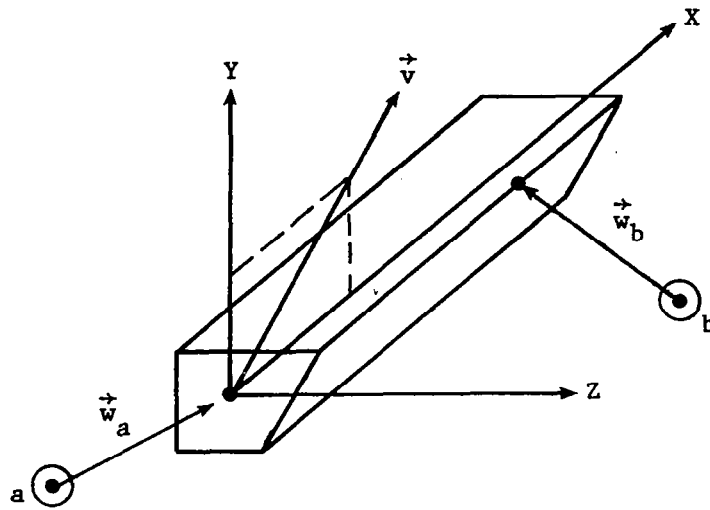
The material property definition cards are used to define the properties for each of the materials used in the structural model. The MAT1 card is used to define the properties for isotropic materials, and it may be referenced by any of the structural elements.

If any two or more elements are truly identical (orientation, geometry, etc.), a CNGRNT card may be used so that the same element matrices are not computed more than once. One element is designated the primary element. The stiffness and mass matrices are calculated for only that element. Other identical elements are designated secondary elements. They use the stiffness and mass matrices previously calculated for the primary element.

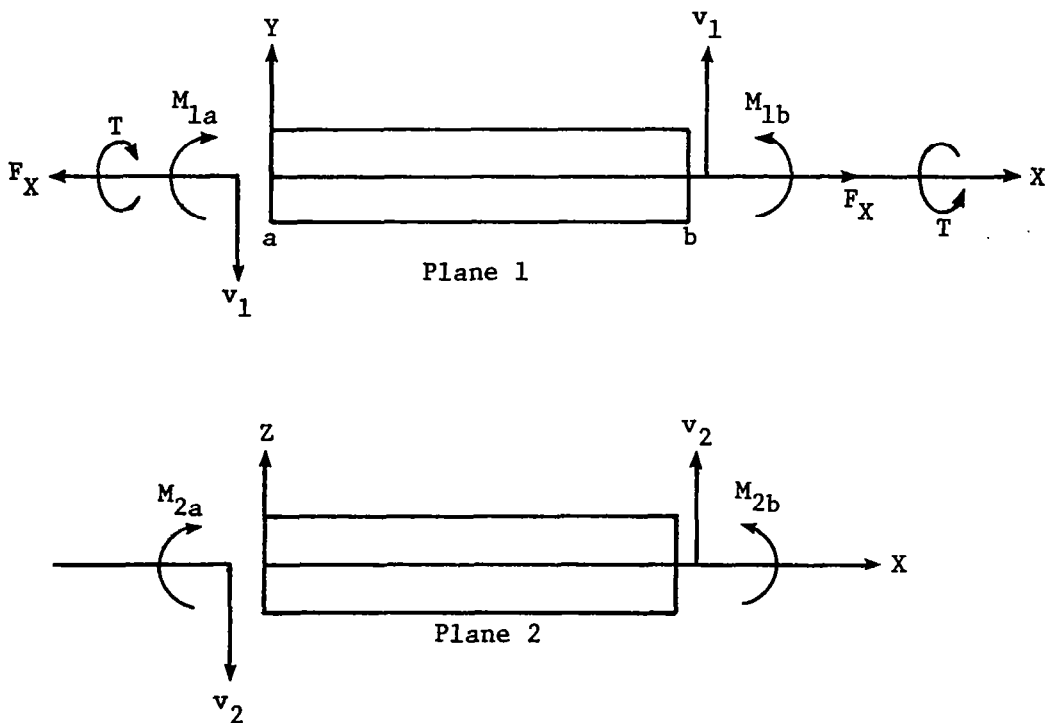
#### 1.4.1 Bar Elements

The bar element is defined with a CBAR card; its properties (constant over the length) are defined with a PBAR card. The bar element includes extension, torsion, bending in two perpendicular planes, and the associated shears. The shear center is assumed to coincide with the elastic axis. Any five of the six forces at either end of the element may be set equal to zero by using the pin flags on the CBAR card. The integers 1 to 6 represent the axial force, the shearing force in plane 1, the shearing force in plane 2, the axial torque, the moment in plane 2, and the moment in plane 1, respectively.

The element coordinate system is shown in figure 4(a). End a is offset from grid point a by an amount measured by vector  $\vec{w}_a$ , and end b is offset from grid point b by an amount measured by vector  $\vec{w}_b$ . The vectors  $\vec{w}_a$  and  $\vec{w}_b$  are measured in the global coordinates of the connected grid point. The X-axis of the element coordinate system is defined by a line connecting ends a and b of the bar element. The orientation of the bar element is described in terms of two reference planes. The reference planes are defined with the aid of vector  $\vec{v}$ . This vector may be defined directly with three components in the global system at end a of the bar or by a line drawn from end a to a third referenced grid point. The first reference plane (plane 1) is defined by the X-axis and the vector  $\vec{v}$ . The second reference plane (plane 2) is defined by the vector cross product ( $\vec{x} \times \vec{v}$ ) and the X-axis. The subscripts 1 and 2 refer to forces and geometric properties associated with bending in planes 1 and 2, respectively. The reference planes are not



(a) Element coordinate system.



(b) Element forces.

Figure 4.- Element coordinate system and element forces for bar.

necessarily principal planes. The coincidence of the reference planes and the principal planes is indicated by a zero product of inertia ( $I_{12}$ ) on the PBAR card. If shearing deformations are included, the reference axes and the principal axes must coincide. When pin flags and offsets are used, the effect of the pin is to free the force at the end of the element X-axis of the beam, not at the grid point. The positive directions for element forces are shown in figure 4(b). The following element forces, either real or complex (depending on the rigid format) are output on request:

Bending moments at both ends in the two reference planes

Shears in the two reference planes

Average axial force

Torque about the bar axis

The following real element stresses are output on request:

Average axial stress

Extensional stress due to bending at four points on the cross section at both ends (optional, calculated only if user enters stress recovery points on PBAR card)

Maximum and minimum extensional stresses at both ends

Margins of safety in tension and compression for the whole element (optional, calculated only if user enters stress limits on MAT1 card)

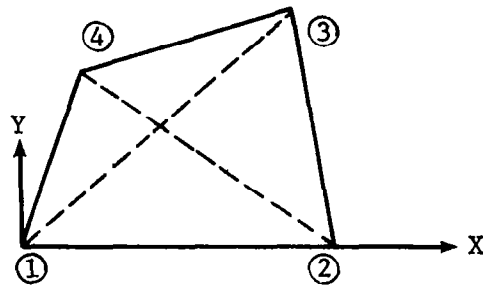
Tensile stresses are given a positive sign, and compressive stresses are given a negative sign. Only the average axial stress and the extensional stresses due to bending are available as complex stresses. The stress recovery coefficients on the PBAR card are used to locate points on the cross section for stress recovery. The subscript 1 is associated with the distance of a stress recovery point from plane 2. The subscript 2 is associated with the distance from plane 1.

#### 1.4.2 Shear Panel Elements

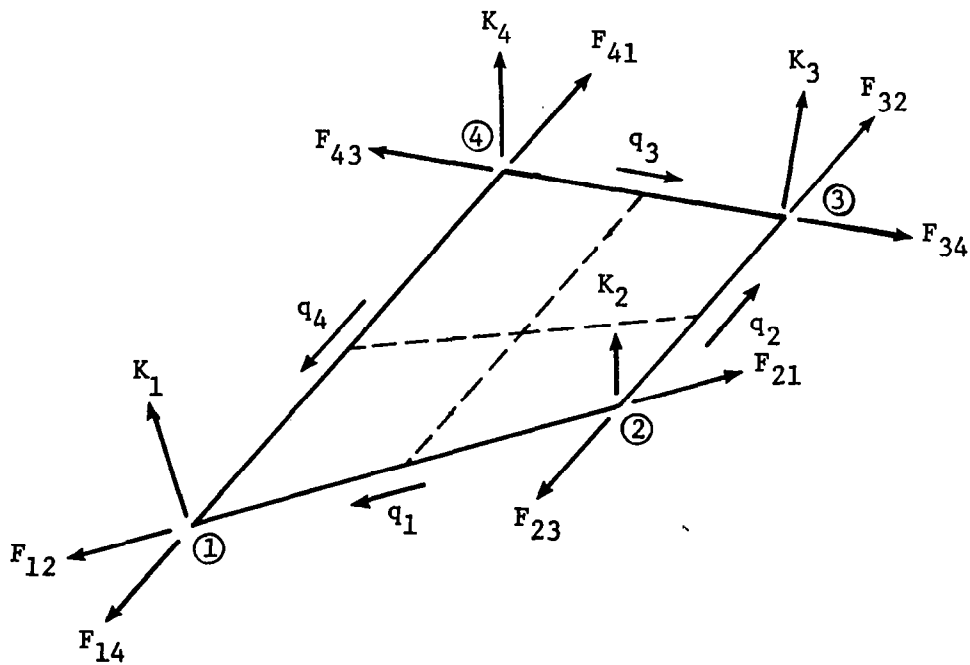
The shear panel is defined with a CSHEAR card, and its properties are defined with a PSHEAR card. A shear panel is a two-dimensional structural element that resists the action of tangential forces applied to its edges but does not resist the action of normal forces. The structural and nonstructural mass of the shear panel is lumped at the connected grid points.

The element coordinate system for a shear panel is shown in figure 5(a). The integers ①, ②, ③, and ④ refer to the order of the connected grid points on the CSHEAR card. The element forces are output on request. The positive directions for these forces are indicated in figure 5(b). These forces consist of the forces applied to the element at the corners in the

direction of the sides, kick forces at the corners in a direction normal to the plane formed by the two adjacent edges, and shear flows (force per unit length) along the four edges. The shear stresses are calculated at the corners in skewed coordinates parallel to the exterior edges. The average of the four corner stresses, the maximum stress, and the margin of safety are output on request.



(a) Coordinate system.



(b) Corner forces and shear flows.

Figure 5.- Coordinate system and element forces for shear panel.

### 1.4.3 Plate Elements

NASTRAN includes triangular plate elements and two different stress systems (membrane and bending) which are uncoupled. There are two different forms of plate elements that are defined by connection cards:

**CTRMEM** - triangular element that has finite inplane stiffness and zero bending stiffness

**CTRIA2** - triangular element that has both inplane stiffness and bending stiffness and assumes a solid homogeneous cross section

The properties for these two elements are defined on the PTRMEM and PTRIA2 cards, respectively. The effect of transverse shear flexibility is automatically included for the CTRIA2 element. Structural mass is based on the membrane thickness. Differential stiffness matrices are generated for the two plate elements.

The element coordinate system for triangular plate and membrane elements is shown in figure 6. The integers 1, 2, and 3 refer to the order of the connected grid points on the connection cards that define the elements.

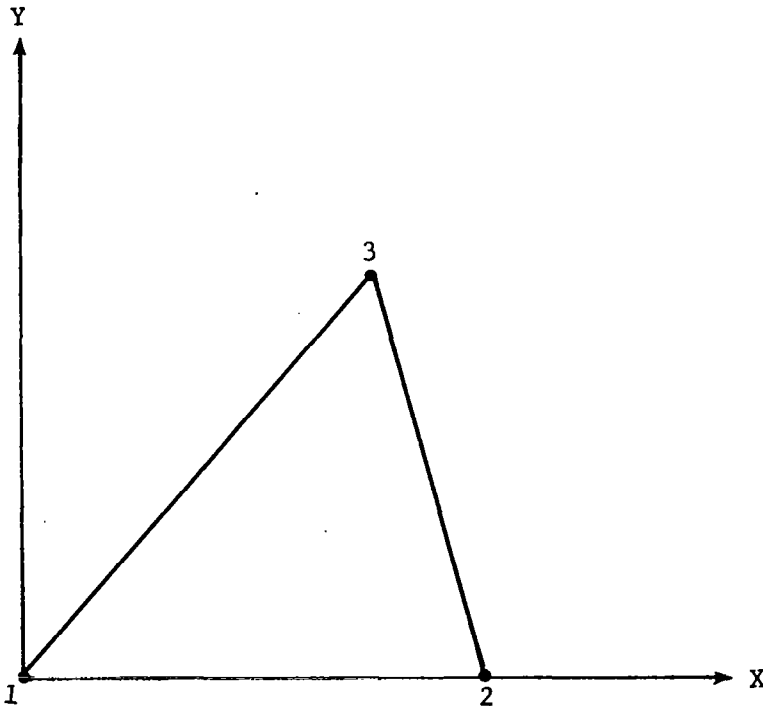


Figure 6.- Plate and membrane element coordinate system.

Average values of element forces are calculated for all plate elements that have a finite bending stiffness. The positive directions for plate and membrane elements, forces, and stresses in the element coordinate system are shown in figure 7. The following element forces per unit of length, either real or complex, are output on request:

Bending moments on the X and Y faces

Twisting moment

Shear forces on the X and Y faces

The following real membrane stresses are output on request:

Normal stresses in the X and Y directions

Shear stress on the X face in the Y direction

Angle between the X-axis and the major principal axis

Major and minor principal stresses

Maximum shear stress

Only the normal stresses and shearing stress are available in the complex form.

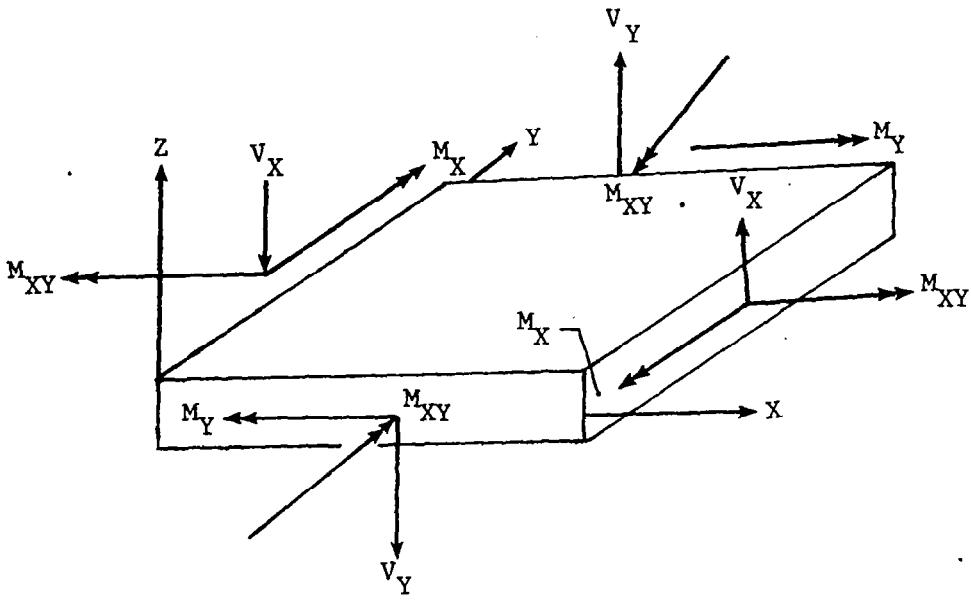
If the plate element has bending stiffness, the average stresses are calculated on the two faces of the plate for homogeneous plates and at two specified points on the cross section for other plate elements. The distances to the specified points are given on the property cards. The positive directions for these fiber distances are defined according to the right-hand sequence of the grid points specified on the connection card. These distances must be non-zero in order to obtain nonzero stress output. The same stresses that are calculated for membrane elements are also calculated for each of the faces.

## 1.5 Mass Element Definitions

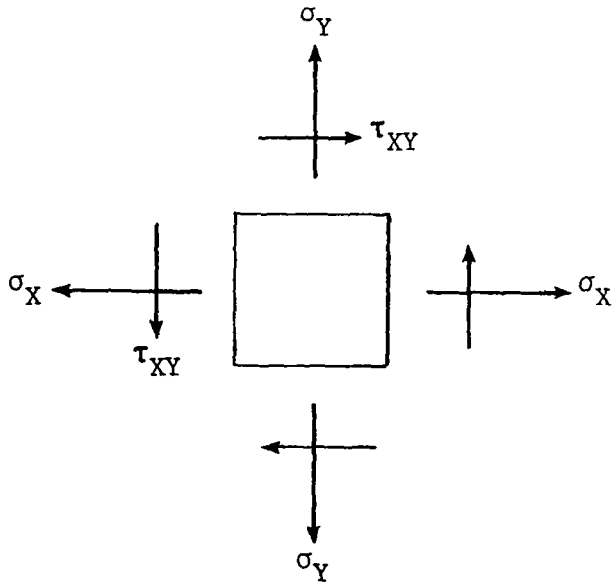
Inertia properties are specified directly as mass elements attached to grid points and indirectly as the properties of stiffness structural elements. In addition, dynamic analysis mass matrix coefficients that are directly referred to the global coordinate system may be specified. Some portions of the mass matrix are generated automatically although other portions are not. Mass data may be assembled according to two different kinds of relationships: lumped mass assumptions or coupled mass considerations.

### 1.5.1 Lumped Mass

The partitions of the lumped mass matrix are shown in equation (1). The terms in this equation are given in the description of the CØNM2 card in section 2.2.



(a) Plate forces and stresses.



(b) Membrane forces and stresses.

Figure 7.- Forces and stresses in plate and membrane elements.

$$M = \begin{bmatrix} \text{Scalar} & \text{1st} \\ \text{1st} & \text{2nd} \\ \text{moment} & \text{moment} \end{bmatrix} \quad (1)$$

The only portion of the lumped mass matrix that is automatically generated is the scalar partition. In this context, automatic generation means calculating the mass from the structural elements that are connected to a given grid point by using only the information provided on the element connection and property card. All the stiffness structural elements (bars, shear panels, and plates) may have uniformly distributed structural and nonstructural mass. Structural mass is calculated from material and geometric properties. The mass is assumed to be concentrated in the middle surface or along the neutral axis for bars so that rotary inertia effects, including the torsional inertia of beams, are absent.

In the lumped mass method, the mass of an element is simply divided into equal portions and each portion is assigned to only one of the surrounding grid points. Thus, for bars, one-half the mass is placed at each end; for uniform triangles, one-third the mass is placed at each corner. The lumped mass matrix is independent of the elastic properties of elements. There are no other automatic routines for providing mass terms for the lumped mass approach.

### 1.5.2 Coupled Mass

In the coupled mass approach, properties of mass that pertain to a single structural element include off-diagonal coefficients that couple action at adjacent grid points. To invoke the automatic generation of the coupled mass matrix, the parameter `CØUPMASS` is indicated on the `PARAM` card. If selected coupled mass properties are desired only for certain element types, a second parameter call is invoked which specifies the desired elements. For further details, see the `PARAM` bulk data card. When using `CØUPMASS`, the nonzero terms are generated in the off-diagonal positions of the mass matrix that correspond generally to nonzero terms of the stiffness matrix. A mass matrix generated by the coupled mass approach, therefore, generally has a density and topology equivalent to that of the stiffness matrix.

Off-diagonal mass terms may also be created during Guyan reduction when the `ØMIT` bulk data card is used to condense the stiffness and mass matrices. Any mass associated with the omitted degrees of freedom is redistributed to the remaining degrees of freedom to form a coupled mass matrix.

### 1.5.3 Mass Input

It may be desired to add mass terms to the structure in addition to those associated with structural elements. For instance, in a lumped mass formulation, additional masses involving rotational degrees of freedom must be independently calculated and input manually by using bulk data cards.



The concentrated mass elements  $C\bar{O}N\bar{M}2$  may be used to add mass terms directly to a single grid point. The  $C\bar{O}N\bar{M}2$  element is used to specify a rigid body with mass and inertia properties that is connected to a single grid point (offsets are allowed).

On the  $C\bar{O}N\bar{M}2$  card, double subscripting is used only for the second moment partition. The program multiplies each cross product of inertia term from  $C\bar{O}N\bar{M}2$  user data by -1 before assembling these data into the mass matrix in order to make them correspond to the requirements of equation (1).

In all cases, a combination of mass input can be used. For instance, the translational inertias can be generated automatically by the element routines, and the first and second moment properties can be provided through  $C\bar{O}N\bar{M}2$  cards. Some elements can be used to provide coupled mass properties through the  $C\bar{O}U\bar{P}M\bar{A}S\bar{S}$  parameter.

#### 1.5.4 Symbols Used in Figures 4, 5, and 7

$F_x$	axial force in bar
$F_{12}, F_{14}, F_{21},$ $F_{23}, F_{32}, F_{34},$ $F_{41}, F_{43}$	inplane corner forces in shear panel
$K_1, K_2, K_3, K_4$	normal or kick forces in shear panels
$M_x, M_y$	direct bending moments in plates
$M_{xy}$	twisting moment in plates
$M_{1a}, M_{1b}, M_{2a}, M_{2b}$	moments in bar
$q_1, q_2, q_3, q_4$	shear flows in shear panel
$T$	torsion in bar
$\vec{v}$	vector defining normal axis
$v_1, v_2$	shear forces in bar
$V_x, V_y$	direct shear forces in plates
$\vec{w}_a, \vec{w}_b$	offset vectors at ends a and b
$\sigma_x, \sigma_y$	direct stresses in plates and membranes
$\tau_{xy}$	shearing stresses in plates and membranes

## 1.6 Constraints and Partitioning

Structural matrices are initially assembled in terms of all structural grid points. These matrices are generated with six degrees of freedom for each geometric grid point and a single degree of freedom for each scalar point. Various constraints are applied to these matrices to remove undesired singularities and to provide boundary conditions and other desired characteristics for the structural model.

Single-point constraints are used to limit a degree of freedom to either zero or a prescribed value. The following three types of bulk data cards are provided for the definition of constraints:

Single-point constraint cards (SPC)

Cards to define reaction points on free bodies (SUPORT)

Cards to define the omitted coordinates in matrix partitioning (ØMIT)

The latter type does not produce constraint forces in static analysis.

### 1.6.1 Single-Point Constraints

A single-point constraint applies a fixed value to a translational or rotational component at a geometric grid point or to a scalar point. One of the most common uses of single-point constraints is to specify the boundary conditions of a structural model by fixing the appropriate degrees of freedom. Multiple sets of single-point constraints can be provided in the bulk data deck, with particular selections made at execution time by using the subcase option in the case control deck, as explained in section 2.3. This option is particularly useful in solving problems that have one or more planes of symmetry.

If elements connected to a grid point do not provide resistance to motion in certain directions, the stiffness matrix is singular. Single-point constraints are used to remove these degrees of freedom from the stiffness matrix. A typical example is a planar structure composed of membrane and extensional elements. The translations normal to the plane and to all three rotational degrees of freedom must be constrained since the corresponding stiffness matrix terms are all zero. If a grid point has a direction of zero stiffness, the single-point constraint need not be aligned exactly in that direction but only needs to have a component in that direction. This allows the use of single-point constraints for the removal of such singularities regardless of the orientation of the global coordinate system. Although the displacements depend on the direction of the constraint, the internal forces are unaffected.

One task performed by the structural matrix assembler is to examine the stiffness matrix for singularities at the grid-point level. Singularities that remain at this level after the application of the single-point constraints are listed in the grid-point singularity table (GPST). The GPST contains all

possible combinations of single-point constraints in the global coordinate system that can be used to remove the singularities.

Single-point constraints are defined on SPC and SPCADD cards. The SPC card is the most general way of specifying single-point constraints. The SPCADD card defines a union of single-point constraint sets specified with the SPC card.

Single-point constraints can also be defined on the GRID card; however, the constraints are part of the model and modifications cannot be made at the subcase level. Also, only zero displacements can be specified on the GRID card.

### 1.6.2 Free-Body Supports

A free body is defined as a structure that is capable of motion without internal stress (i.e., a free body has one or more rigid-body degrees of freedom). The stiffness matrix for a free body is singular, with the defect equal to the number of stress-free or rigid-body modes. A solid three-dimensional body has up to six rigid-body modes. Linkages and mechanisms can have a greater number. To permit the analysis of mechanisms, no restriction is placed in the program on the number of stress-free modes.

Free-body supports are defined with a SUPPORT card. In either case, only a single set can be specified, and if such cards appear in the bulk data deck, they are automatically used in the solution. Free-body supports must be defined in the global coordinate system.

In static analysis by the displacement method, rigid-body modes must be restrained to remove the singularity of the stiffness matrix. The required constraints may be supplied with single-point constraints or free-body supports. If free-body supports are used, the rigid-body characteristics are calculated and a check is made on the sufficiency of the supports. Such a check is obtained by calculating the rigid-body error ratio as defined in the rigid-body matrix generator operation. This error ratio is automatically printed following the execution of the rigid-body matrix generator. The error ratio should be zero but may be nonzero for any of the following reasons:

- Round-off error accumulation

- Insufficient free-body supports

- Redundant free-body supports

The redundancy of the supports may be caused by improper use of the free-body supports themselves or by the presence of single-point constraints that limit the rigid-body motions.

Rigid-body vibration modes are calculated by a separate procedure, provided that a set of free-body supports are supplied by the user. This process improves efficiency and, sometimes, reliability. If the user does not specify free-body supports (or specifies an insufficient number of supports) the (remaining) rigid-body modes are calculated by the method selected for the

finite frequency modes, provided zero frequency is included in the range of interest. If the user does not provide free-body supports and if zero frequency is not included in the range of interest, the rigid-body modes are not calculated.

### 1.6.3 Partitioning

A two-way partitioning scheme is provided as an optional feature for the NASTRAN model. The partitions are defined by listing the degrees of freedom for one of the partitions on the OMIT card. These degrees of freedom are referred to as the omitted set. The remaining degrees of freedom are referred to as the analysis set.

Partitioning can be used to improve efficiency in solving ordinary statics problems where the bandwidth of the unpartitioned stiffness matrix is large enough to cause excessive use of secondary storage devices during the triangular decomposition of the stiffness matrix. In this application, the analysis set should be relatively small and should be selected so that the omitted set consists of uncoupled partitions, each having a bandwidth of approximately the same size but being smaller than the original matrix. The omitted set might be thought of as consisting of several substructures that are coupled to the analysis set.

One of the more important applications of partitioning is Guyan reduction. This technique is a means for reducing the number of degrees of freedom used in vibration analysis with a minimum loss of accuracy. Guyan reduction is based on the fact that considerably fewer grid points are needed to describe the inertia of a structure than are needed to describe its elasticity with comparable accuracy. The error in the approximation is small, provided that the set of displacements used for vibration analysis is judiciously chosen. The set's members should be uniformly dispersed throughout the structure, and all large mass items should be connected to grid points that are members of the analysis set.

The user is cautioned to consider that the matrix operations associated with this partitioning procedure tend to create nonzero terms and to fill what were previously very sparse matrices. The partitioning option is most effectively used if the members of the omitted set are either a very large fraction or a very small fraction of the total set. In most of the applications, the omitted set is a large fraction of the total set, and the matrices used for analysis, although small, are usually full. If the analysis set is not a small fraction of the total set, a solution that uses the larger but sparser matrices may well be more efficient. The partitioning option can also be used to make modest reductions in the order of the problem by placing a few scattered grid points in the omitted set. If the points in the omitted set are uncoupled, the sparseness in the matrices is well preserved.

### 1.7 Static Loads

In NASTRAN, static loads are applied to grid points in two ways:

Loads applied directly to grid points (using FORCE and MOMENT cards)

Equivalent loads resulting from enforced displacements of grid points (using SPC cards)

Any number of load sets can be defined in the bulk data deck. However, only those sets selected in the case control deck as described in section 2.3 are used in the problem solution. The manner of selecting each type of load is specified on the associated bulk data card described in section 2.2.

The FORCE card is used to define a static load applied to a geometric grid point in terms of components defined by a local coordinate system. The orientation of the load components depends on the type of local coordinate system used to define the load. The directions of the load components are the same as those indicated in figure 1 for displacement components. The MOMENT cards are used in a similar fashion to define the application of a concentrated moment at a grid point.

Equivalent loads resulting from enforced displacements of grid points are calculated by the program and are added to the other applied loads. The magnitudes of the enforced displacements are specified on SPC cards in the global coordinate system. The application of the load is automatic when the user selects the associated SPC set in the case control deck.

The LOAD card in the bulk data deck defines a static loading condition that is a linear combination of load sets which consist of loads applied directly to grid points. The application of the combined loading condition is requested in the case control deck by selecting the set number of the LOAD combination.

Equivalent loads (enforced deformation and enforced displacement) must have unique set identification numbers and be separately selected in the case control deck. For any particular solution, the total static load is the sum of the applied loads and the equivalent loads.

## 1.8 Analyses Descriptions

This condensed form of NASTRAN is merely a limited capability form of NASTRAN Level 16. In general, it can be used to solve three types of problems: statics, vibration, and buckling. A description of each type analysis follows. The user should refer to the appendix to see how to apply a DMAP sequence to static, vibration, and buckling analyses in particular. In general, DMAP may be used for any problem.

### 1.8.1 Linear Static Analysis

The linear static analysis uses most of the features available in NASTRAN for model definition. These features include specification of coordinate

systems, grid-point locations, element connectivities, material properties, and constraints, as well as direct grid-point and element loadings. The results available for output consist of weight and balance information, displacements, grid-point loadings, element forces and stresses, and support reactions. The user may also request plots of both the undeformed and the deformed structural model.

### 1.8.2 Vibration Analysis

This method provides for a normal modes analysis of undamped systems with symmetric matrices. It generates both the mass and stiffness matrices and also offers the inverse power method for eigenvalue extraction. Available output includes the resulting eigenvalues and natural frequencies, mode shape deflections, modal forces and stresses for selected elements, modal reaction forces, modal mass, and plots of the structural model for each mode shape.

### 1.8.3 Buckling Analysis

NASTRAN provides for differential stiffness. The differential stiffness effects provide the user with a second-order approximation to the nonlinear effects of large deflections. Applied loads are assumed to move with their points of application and remain fixed in magnitude and direction. The differential stiffness matrix is computed as an addition to the original stiffness matrix from the work done by each element in response to a user-specified preload. Buckling analysis takes the differential stiffness matrix and, together with the original stiffness matrix, solves an eigenvalue problem to obtain the critical load factor that will produce buckling. This critical load factor, or eigenvalue, multiplied into the preload vector gives the critical loading at which the structure would go unstable. The eigenvector obtained is the mode shape in which the structure would buckle.

### 1.8.4 Direct Matrix Abstraction Programs

The user also has the DMAP (direct matrix abstraction program) programming language at his disposal. The DMAP facilities include the instructions that are required to perform matrix operations and data manipulation, to exercise executive control, to perform structural computations, to exercise user-designed functions programmed by the user, and to output the desired results. Utility DMAP instructions may be used to output data to tape for input to programs external to NASTRAN and to read input data generated outside of NASTRAN. When difficult modeling problems arise, these DMAP instructions may be used to retrieve internal NASTRAN data blocks, to perform special operations to verify the accuracy of a solution, or to extract intermediate results. All these options can be used to help detect the source of errors, to avoid problems, or to improve the results. This extensive capability allows the user to go beyond the scope of the existing analyses and solve a broad range of nonstructural matrix problems within the framework of NASTRAN.

## 2. NASTRAN DATA DECK

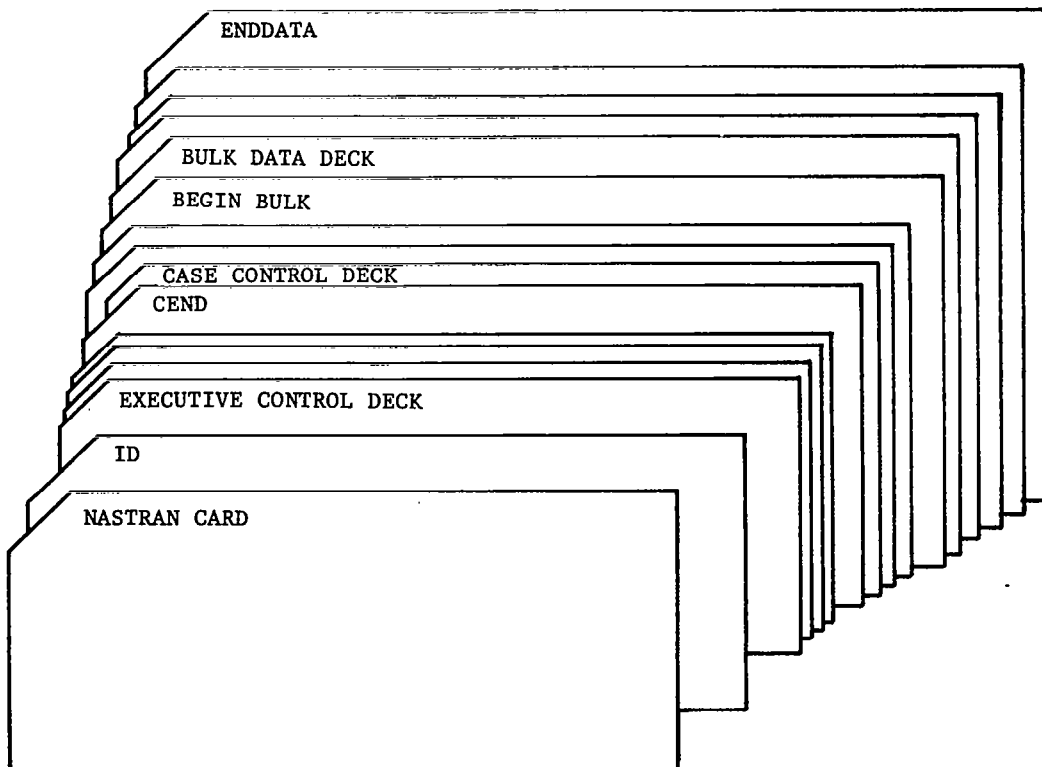
### 2.1 General Description of NASTRAN Data Deck

The NASTRAN data deck is constructed in the following order:

- NASTRAN Card (section 2.5)
- Executive Control Deck (section 2.4)
- Case Control Deck (section 2.3)
- Bulk Data Deck (section 2.2)

The NASTRAN card is optional but if used must always come first. It is used to change the default values for certain system parameters. The NASTRAN card is followed by the executive control deck, which identifies the job and the type of solution to be performed and also contains DMAP. The case control deck follows. This deck selects items from the bulk data deck, which comes last. The bulk data cards are used to define the structural model and the various pools of data which may be selected by case control at execution time. The bulk data deck is usually the first group of data prepared by a user and hence is described first in this guide.

The general construction of the NASTRAN data deck is shown as follows:

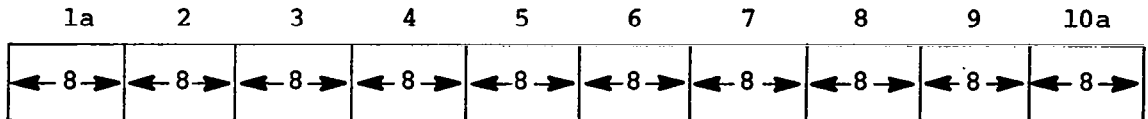


## 2.2 Bulk Data Deck

The primary NASTRAN input medium is the bulk data card. These cards are used to define the structural model and the various pools of data which may be selected by case control at execution time.

The bulk data deck may be submitted with the cards in any order because a sort is performed prior to the execution of the input file processor. The user may obtain a printed copy of either the unsorted or sorted bulk data by selection in the case control deck.

The bulk data card format is variable to the extent that any quantity except the mnemonic can be punched anywhere within a specified 8- or 16-column field. The normal card uses an 8-column field, as indicated in the following diagram of a small field bulk data card:



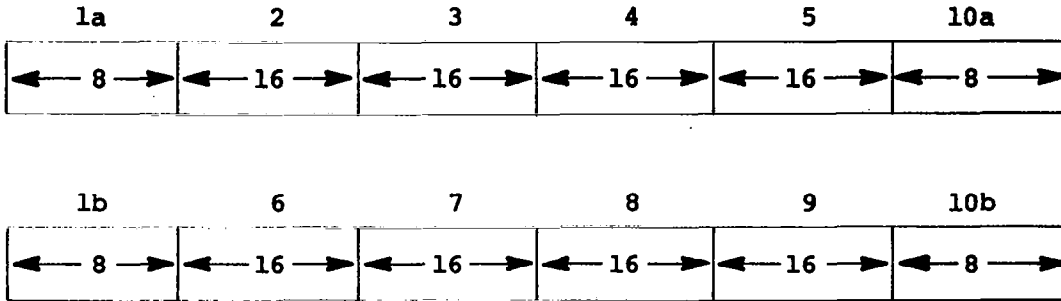
The mnemonic is punched in field 1 beginning in column 1. Fields 2 to 9 are for data items. The only limitations in data items are that they must lie completely within the designated field, have no embedded blanks, and be of the proper type (i.e., blank, integer, real, double precision, or BCD). All real numbers including zero must contain a decimal point. A blank will be interpreted as a real zero or integer zero as required. Real numbers may be encoded in various ways. For example, the real number 7.0 may be encoded as 7.0, .7E1, 0.7+1, 70.-1, or .70+1. A double-precision number must contain both a decimal point and an exponent with the character D, such as 7.0D0. Double-precision data values are only allowed in a few situations, such as on the PARAM card. BCD data values consist of one to eight alphanumeric characters, the first of which must be alphabetic.

Normally field 10 is reserved for optional user identification. However, for continuation cards, field 10 (except column 73 which is not referenced) is used in conjunction with field 1 of the continuation card as an identifier and hence must contain a unique entry. The continuation card contains the symbol + in column 1, followed by the same seven characters that appeared in columns 74 to 80 of field 10 of the card that is being continued. This arrangement allows the data to be submitted as an unsorted deck.

The small field data card should be more than adequate for the kinds of data normally associated with structural engineering problems. Since abbreviated forms of floating point numbers are allowed, up to seven significant



decimal digits may be used in an eight-character field. Occasionally, however, the input is generated by another computer program or is available in a form where a wider field would be desirable; then, the larger field format with a 16-character data field is provided. Each logical card consists of two physical cards, as indicated in the following diagram for a large field bulk data card:



The large field card is denoted by placing the symbol \* after the mnemonic in field 1a and some unique character configuration in the last seven columns of field 10a. The second physical card contains the symbol \* in column 1 followed by the same seven characters that appeared after column 73 in field 10a of the first card. The second card may in turn be used to point to a large or small field continuation card, depending on whether the continuation card contains the symbol \* or the symbol + in column 1. The use of multiple and large field cards is illustrated in the following examples:

**Small Field Card With Small Field Continuation Card**

TYPE										QED123
+ED123										

**Large Field Card**

TYPE*										QED124
*ED124										

Large Field Card With Large Field Continuation Card

TYPE*						QED301
*ED301						QED302
*ED302						QED305
*ED305						

Large Field Card Followed by a Small Field Continuation Card and a Large Field Continuation Card

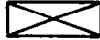
TYPE*						QED462
*ED462						QED421
+ED421						QED361
*ED361						QED291
*ED291						

Small Field Card With Large Field Continuation Card

TYPE							QED632
*ED632							QED204
*ED204							

In the above examples, column 73 arbitrarily contains the symbol Q whenever field 10 is used as a pointer. However, column 73 could have been left blank or contained the same symbol used in column 1 of the following card (i.e., the symbols \* or +).

The detailed descriptions of the bulk data cards are contained in this section in alphabetical order. Small field examples are given for each card along with a description of the contents of each field. In the format and example section of each card description, both a symbolic card format description and an example of an actual card are shown. Literal constants are shown in the card format section enclosed in quotes (e.g., "0"). Fields that must

necessarily be blank are indicated in the card format section by  whenever they are followed by nonblank fields or whenever such notation will clarify the card description.

The input file processor produces error messages for any cards that do not have the proper format or that contain illegal data.

Continuation cards need not be present unless they contain required data. For multiple continuation cards, the intermediate cards must be present (even though fields 2 to 9 are blank) if one of the following cards contains data in fields 2 to 9. In addition, a double-field format requires at least two cards (or subsequent multiples of two) so that ten data fields are included. Thus one or more double-field cards may contain no data.

The bulk data cards used in the condensed form of NASTRAN are as follows:

CBAR	simple beam element connection
CNGRNT	identical elements indicator
<del>C</del> ONM2	concentrated mass element connection
<del>C</del> ORD1C	cylindrical coordinate system definition
<del>C</del> ORD1R	rectangular coordinate system definition
<del>C</del> ORD1S	spherical coordinate system definition
CSHEAR	shear panel element connection
CTRIA2	triangular element connection with bending
CTRMEM	triangular element connection
DMI	direct matrix input
EIGB	buckling analysis data
EIGR	real eigenvalue extraction data
<del>F</del> ORCE	static load
GRID	grid point
<del>L</del> OAD	static load combination
MAT1	material property definition
<del>M</del> OMENT	static moment

ØMIT	omitted coordinates
PARAM	parameter
PBAR	simple beam property
PSHEAR	shear panel property
PTRIA2	homogeneous triangular element property
PTRMEM	triangular membrane property
SPC	single-point constraint
SPCADD	union of single-point constraint sets
SUPØRT	fictitious support
\$	comment

Input Data Card CBAR Simple Beam Element Connection

Description: Defines a simple beam element (BAR) of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CBAR	EID	PID	GA	GB	X1,GØ	X2	X3	F	abc
CBAR	2	39	7	3	13.	14.	15.	2	123
+bc	PA	PB	Z1A	Z2A	Z3A	Z1B	Z2B	Z3B	
+23		513							

<u>Field</u>	<u>Contents</u>
EID	Unique element identification number (Integer > 0)
PID	Identification number of a PBAR property card; default is EID (Integer > 0)
GA,GB	Grid-point identification numbers of connection points (Integer > 0, GA ≠ GB)
X1,X2,X3	Components of vector $\vec{v}$ at end a (fig. 1(a)) measured at end a; parallel to the components of the displacement coordinate system for GA; used to determine (with the vector from end a to end b) the orientation of the element coordinate system for the bar element (real; $X1^2 + X2^2 + X3^2 > 0$ )
GØ	Grid-point identification number to optionally supply X1, X2, and X3 (Integer > 0, see diagram under field F)
F	Flag to specify the nature of fields 6 to 8 as follows:

	6	7	8
F = blank			
F = 1	X1	X2	X3
F = 2	GØ	blank/0	blank/0

PA, PB

Pin flags for bar ends a and b, respectively, that are used to insure that the bar cannot resist a force or moment corresponding to the pin flag at that respective end of the bar. Up to five of the unique digits 1 to 6 are allowed anywhere in the field with no embedded blanks (Integer > 0). These degree-of-freedom codes refer to element forces and not global forces. The bar must have stiffness associated with the pin flag. For example, if pin flag 4 is specified, the bar must have a value for J, the torsional constant.

Z1A, Z2A, Z3A  
Z1B, Z2B, Z3B

Components of offset vectors  $\vec{w}_a$  and  $\vec{w}_b$ , respectively (fig. 4(a)) in displacement coordinate systems at points GA and GB, respectively (real or blank)

- Remarks:
1. Element identification numbers must be unique with respect to all other element identification numbers.
  2. If there are no pin flags or offsets, the continuation card may be omitted.

Input Data Card CNGRNT Identical Elements Indicator

Description: Designates secondary element(s) identical to a primary element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CNGRNT	PRID	SECID1	SECID2	SECID3	SECID4	SECID5	SECID6	SECID7	abc
CNGRNT	11	2	17	34	35	36			
+bc	SECID8	SECID9		etc.					

Alternate Form

CNGRNT	PRID	SECID1	"THRU"	SECID2					
CNGRNT	7	2	THRU	55					

Field

Contents

**PRID** Identification number of the primary element (not necessarily the lowest number) for which the stiffness and mass matrices will be calculated

**SECIDi** Identification number(s) of secondary element(s) whose matrices will be identical to the primary element

Remarks: 1. Orientation, geometry, etc., must be truly identical so that the same stiffness and mass matrices are generated in the global coordinate system.


2. An element that has been listed as a primary element on any CNGRNT card cannot be listed as a secondary element either on that card or on any other CNGRNT card.

Input Data Card CØNM2

Concentrated Mass Element Connection

Description: Defines a concentrated mass at a grid point of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CØNM2	EID	G	CID	M	X1	X2	X3		abc
CØNM2	2	15	6	49.7					123
+bc	I11	I21	I22	I31	I32	I33			
+23	16.2		16.2			7.8			

Field

Contents

- EID           Element identification number (Integer > 0)
- G             Grid-point identification number (Integer > 0)
- CID           Coordinate system identification number (Integer ≥ 0)
- M             Mass value (real)
- X1,X2,X3     Offset distances for the mass in the coordinate system defined in field 4 (real)
- Iij           Mass moments of inertia measured at the mass center of gravity in the coordinate system defined by field 4 (real)

- Remarks:
1. Element identification numbers must be unique with respect to all other element identification numbers.
  2. The continuation card may be omitted.



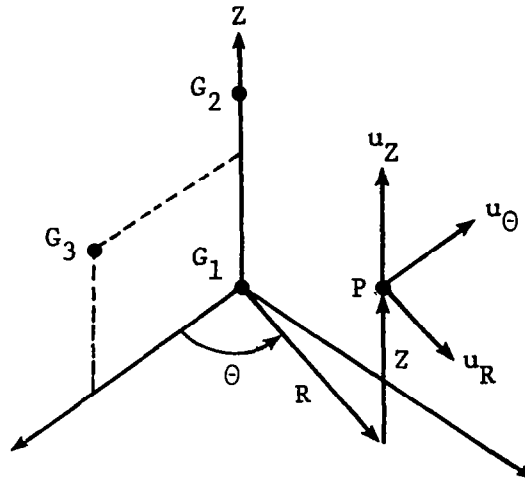
3. The form of the inertia matrix about its center of gravity is taken as follows:

$$\begin{bmatrix}
 M & & & 0 & zM & -yM \\
 & M & & -zM & 0 & xM \\
 & & M & yM & -xM & 0 \\
 & & & I_{11} & -I_{21} & -I_{31} \\
 & \text{SYM.} & & & I_{22} & -I_{32} \\
 & & & & & I_{33}
 \end{bmatrix}$$

Input Data Card CØRD1C

Cylindrical Coordinate System Definition

Description: Defines a cylindrical coordinate system by reference to three grid points. These points must be defined in coordinate systems whose definition does not involve the coordinate system being defined. The first point G1 is the origin, the second G2 lies on the Z-axis, and the third G3 lies in the plane of the azimuthal origin:



Format and Example:

1	2 3 4 5				6 7 8 9				10
CØRD1C	CID	G1	G2	G3	CID	G1	G2	G3	
CØRD1C	3	16	32	19					

Field

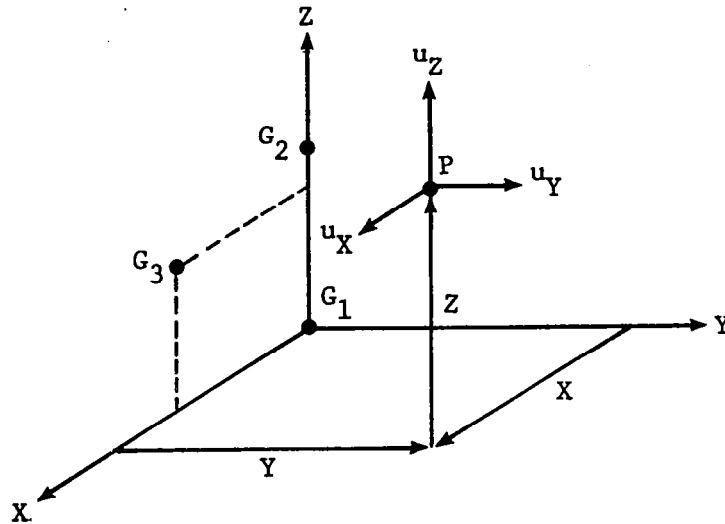
Contents

- CID                    Coordinate system identification number (Integer > 0)
- G1,G2,G3            Grid-point identification numbers (Integer > 0; G1 ≠ G2 ≠ G3)

- Remarks:
1. Coordinate system identification numbers on all CØRD1R, CØRD1C, and CØRD1S cards must be unique.
  2. The three points G1, G2, and G3 must be noncolinear.

3. The location of a grid point (P in the sketch) in this coordinate system is given by  $(R, \theta, Z)$  where  $\theta$  is measured in degrees.
4. The displacement coordinate directions at P are dependent on the location of P, as shown by  $(u_R, u_\theta, u_Z)$ .
5. Points on the Z-axis may not have their displacement directions defined in this coordinate system since an ambiguity results.
6. One or two coordinate systems may be defined on a single card.

**Description:** Defines a rectangular coordinate system by reference to three grid points. These points must be defined in coordinate systems whose definition does not involve the coordinate system being defined. The first point G1 is the origin, the second G2 lies on the Z-axis, and the third G3 lies in the X-Z plane.



**Format and Example:**

1	2 3 4 5				6 7 8 9				10
CØRD1R	CID	G1	G2	G3	CID	G1	G2	G3	
CØRD1R	3	16	32	19					

**Field**

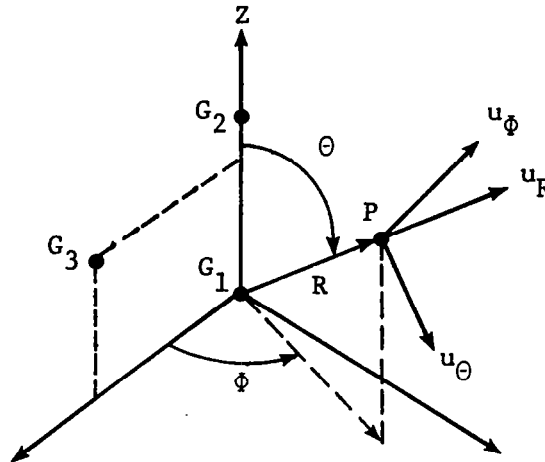
**Contents**

- CID                      Coordinate system identification number (Integer > 0)
- G1,G2,G3                Grid-point identification numbers (Integer > 0; G1 ≠ G2 ≠ G3)

- Remarks:**
1. Coordinate system identification numbers on all CØRD1R, CØRD1C, and CØRD1S cards must be unique.
  2. The three points G1, G2, and G3 must be noncolinear.

3. The location of a grid point (P in the sketch) in this coordinate system is given by  $(X, Y, Z)$ .
4. The displacement coordinate directions at P are shown by  $(u_x, u_y, u_z)$ .
5. One or two coordinate systems may be defined on a single card.

Description: Defines a spherical coordinate system by reference to three grid points. These points must be defined in coordinate systems whose definition does not involve the coordinate system being defined. The first point G1 is the origin, the second G2 lies on the Z-axis, and the third G3 lies in the plane of the azimuthal origin.



Format and Example:

1	┌───────────┐				┌───────────┐				10
2	3	4	5	6	7	8	9		
<u>CØRD1S</u>	CID	G1	G2	G3	CID	G1	G2	G3	
<u>CØRD1S</u>	3	16	32	19					

Field

Contents

- CID            Coordinate system identification number (Integer > 0)
- G1,G2,G3      Grid-point identification numbers (Integer > 0; G1 ≠ G2 ≠ G3)

- Remarks:
1. Coordinate system identification numbers on all CØRD1R, CØRD1C, and CØRD1S cards must be unique.
  2. The three points G1, G2, and G3 must be noncolinear.

3. The location of a grid point (P in the sketch) in this coordinate system is given by  $(R, \theta, \phi)$  where  $\theta$  and  $\phi$  are measured in degrees.
4. The displacement coordinate directions at P are dependent on the location of P, as shown by  $(u_R, u_\theta, u_\phi)$ .
5. Points on the polar axis may not have their displacement directions defined in this coordinate system since an ambiguity results.
6. One or two coordinate systems may be defined on a single card.

Input Data Card CSHEAR Shear Panel Element Connection

Description: Defines a shear panel element (SHEAR) of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CSHEAR	EID	PID	G1	G2	G3	G4			
CSHEAR	3	6	1	5	3	7			

Field

Contents

- EID Element identification number (Integer > 0)
- PID Identification number of a PSHEAR property card; default is EID (Integer > 0)
- G1,G2,G3,G4 Grid-point identification numbers of connection points (Integer > 0; G1 ≠ G2 ≠ G3 ≠ G4)

- Remarks:
1. Element identification numbers must be unique with respect to all other element identification numbers.
  2. Grid points G1 to G4 must be ordered consecutively around the perimeter of the element.
  3. All interior angles must be less than 180°.



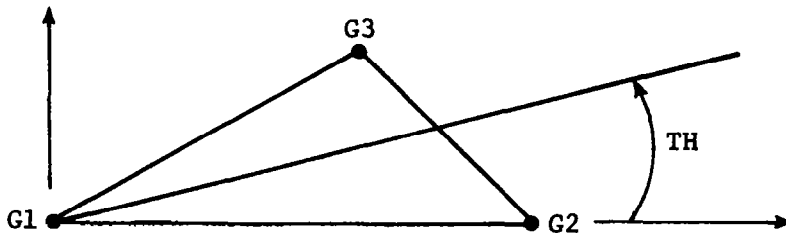
Input Data Card CTRIA2 Triangular Element Connection

Description: Defines a triangular membrane and bending element (TRIA2) of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CTRIA2	EID	PID	G1	G2	G3	TH			
CTRIA2	16	2	12	1	3	16.2			

<u>Field</u>	<u>Contents</u>
EID	Element identification number (Integer > 0)
PID	Identification number of a PTRIA2 property card; default is EID (Integer > 0)
G1,G2,G3	Grid-point identification numbers of connection points (Integer > 0; G1 ≠ G2 ≠ G3)
TH	Material property orientation angle in degrees (real). The following sketch gives the sign convention for TH.



- Remarks:
1. Element identification numbers must be unique with respect to all other element identification numbers.
  2. Interior angles must be less than 180°.

Input Data Card CTRMEM Triangular Element Connection

Description: Defines a triangular membrane element (TRMEM) of the structural model.

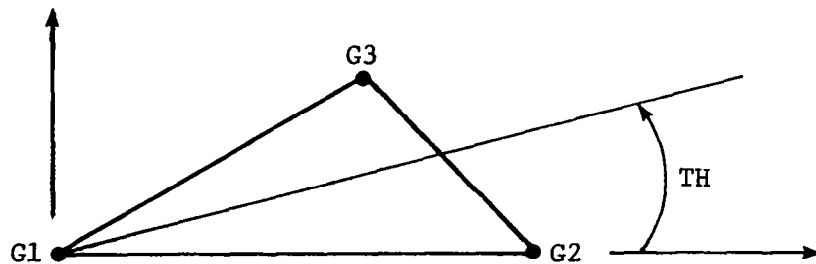
Format and Example:

1	2	3	4	5	6	7	8	9	10
CTRMEM	EID	PID	G1	G2	G3	TH			
CTRMEM	16	2	12	1	3	16.3			

Field

Contents

- EID Element identification number (Integer > 0)
- PID Identification number of a PTRMEM property card; default is EID (Integer > 0)
- G1,G2,G3 Grid-point identification numbers of connection points (Integer > 0; G1 ≠ G2 ≠ G3)
- TH Material property orientation angle in degrees (real). The following sketch gives the sign convention for TH.



- Remarks:
1. Element identification numbers must be unique with respect to all other element identification numbers.
  2. Interior angles must be less than 180°.


Input Data Card DMI Direct Matrix Input

Description: Used to define the matrix data blocks directly. Generates a matrix of the form

$$[A] = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ A_{m1} & \cdot & \dots & A_{mn} \end{bmatrix}$$

where the elements  $A_{ij}$  may be real or complex single-precision or double-precision numbers.

Formats and Example: (The first logical card is a header card.)

1	2	3	4	5	6	7	8	9	10
DMI	NAME	"0"	FØRM	TIN	TØUT		M	N	
DMI	QQQ	0	2	3	3		4	2	
DMI	NAME	J	I1	A(I1,J)	A(I1+1,J)		etc.	I2	+abc
DMI	QQQ	1	1	1.0	2.0	3.0	4.0	3	+1
+abc	A(I2,J)		etc.						
+1	5.0	6.0							
DMI	QQQ	2	2	6.0	7.0	4	8.0	9.0	

This same format is used for each nonnull column.

<u>Field</u>	<u>Contents</u>
NAME	Any NASTRAN BCD value (1 to 8 alphanumeric characters, the first of which must be alphabetic) which will be used in the DMAP sequence to reference the data block

FØRM

Code for type of matrix:

- 1 square matrix (not symmetric)
- 2 general rectangular matrix
- 6 symmetric matrix

TIN

Type of matrix being input as follows:

- 1 real, single precision (one field is used per element)
- 2 real, double precision (one field is used per element)
- 3 complex, single precision (two fields are used per element)
- 4 complex, double precision (two fields are used per element)

TØUT

Type of matrix which will be created:

- 1 real, single precision
- 2 real, double precision
- 3 complex, single precision
- 4 complex, double precision

M

Number of rows in A (Integer > 0)

N

Number of columns in A (Integer > 0)

J

Column number of A (Integer > 0)

I1, I2, etc.

Row number of A (Integer > 0)

A(Ix, J)

Element of A (real), (see TIN)

Remarks: 1. The user must write a DMAP (or make alterations to a rigid format) to use the DMI feature since a data block is being defined. All rules governing the use of data blocks in DMAP sequences apply. In the example previously shown, the data block QQQ is defined as a complex, single-precision, rectangular 4 × 2 matrix.

$$[QQQ] = \begin{bmatrix} (1.0, 2.0) & (0.0, 0.0) \\ (3.0, 4.0) & (6.0, 7.0) \\ (5.0, 6.0) & (0.0, 0.0) \\ (0.0, 0.0) & (8.0, 9.0) \end{bmatrix}$$

The DMAP data block NAME (QQQ in the example) appears in the initial FIAT and the data block initially appears on the data pool file (PØØL).

2. A limit to the number of DMI's which may be defined is set by the size of the data pool dictionary. The total number of DMI's may not exceed this size.
3. There are a number of reserved words which may not be used for DMI names. Among these are PØØL, NPTP, ØPTP, UMF, NUMF, PLT1, PLT2, INPT, GEØM1, GEØM2, GEØM3, GEØM4, GEØM5, EDT, MPT, DPT, DIT, DYNAMICS, IFFFILE, AXIC, FØRCE, MATPØØL, PCDB, XYCDB, CASECC, any DTI names, and SCRATCH1 to SCRATCH9.
4. Field 3 of the header card must contain an integer 0.
5. For symmetric matrices, the entire matrix must be input.
6. Only nonzero terms need be entered.
7. A blank field on this card is not equivalent to a zero. If zero input is desired, the appropriate type zero must be punched (i.e., 0.0 or 0.0D0).
8. Complex input must have both the real and imaginary parts punched if either part is nonzero.
9. A new column requires that a new card be started.

Input Data Card EIGB Buckling Analysis Data

Description: Defines the data needed to perform buckling analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
EIGB	SID	METHØD	L1	L2	NEP	NDP	NDN	E	+abc
EIGB	13	INV	0.1	2.5	2	1	1	0.0	ABC
+abc	NØRM	G	C						
+BC	MAX								

Field

Contents

- SID Set identification number (Unique integer > 0)
- METHØD Method of eigenvalue extraction, INV  
 INV - Inverse power method; symmetric matrix operations
- L1,L2 Eigenvalue range of interest (real; L1 < L2 > 0.0)
- NEP Estimate of number of roots in positive range (Integer > 0)
- NDP,NDN Desired number of positive and negative roots; Default = 3 NEP (Integer > 0)
- E Convergence criteria, optional (Real ≥ 0.0)
- NØRM Method for normalizing eigenvectors; one of the BCD values MAX or PØINT  
 MAX - Normalize to unit value of the largest component in the analysis set  
 PØINT - Normalize to unit value of the component defined in fields 3 and 4; defaults to MAX if defined component is zero
- G Grid- or scalar-point identification number; required if and only if NØRM = PØINT (Integer > 0)
- C Component number; required if and only if NØRM = PØINT and G is a geometric grid point (one of the integers 1 to 6)

- Remarks:
1. Buckling analysis root extraction data sets must be selected in the case control deck (METHOD = SID) to be used by NASTRAN.
  2. The quantities L1 and L2 are dimensionless and specify a range in which the eigenvalues are to be found. An eigenvalue is a factor by which the prebuckling state of stress (first subcase) is multiplied to produce buckling.
  3. The continuation card is required.
  4. If NORM = MAX, components that are not in the analysis set may have values larger than unity.
  5. If NORM = POINT, the selected component must be in the analysis set.

Input Data Card EIGR Real Eigenvalue Extraction Data

Description: Defines the data needed to perform real eigenvalue analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
EIGR	SID	METHØD	F1	F2	NE	ND	NZ	E	+abc
EIGR	13	INV	1.9	15.6	10	12	0	1.-3	ABC
+abc	NØRM	G	C						
+BC	PØINT	32	4						

Field

Contents

SID Set identification number (Unique integer > 0)

METHØD Method of eigenvalue extraction, INV  
 INV - Inverse power method; symmetric matrix operations

F1,F2 Frequency range of interest; required for METHØD = INV  
 (Real  $\geq$  0.0; F1 < F2). The frequency range is ignored if  
 ND > 0, in which case the eigenvectors for the first ND  
 positive roots are found (real; F1 < F2)

NE Estimate of number of roots in range; required for METHØD = INV  
 (Integer > 0)

ND Desired number of roots for METHØD = INV; default is 3 NE  
 (Integer > 0)

NZ Number of free-body modes (not used for METHØD INV)

E Mass orthogonality test parameter; default is 0.0, which means  
 no test will be made (Real  $\geq$  0.0)



**NØRM** Method for normalizing test parameter; default is 0.0, which means no test will be made (Real  $\geq$  0.0)

**MASS** - Normalize to unit value of the generalized mass

**MAX** - Normalize to unit value of the largest component in the analysis set

**PØINT** - Normalize to unit value of the component defined in fields 3 and 4; defaults to **MAX** if defined component is zero

**G** Grid- or scalar-point identification number; required if and only if **NØRM** = **PØINT** (Integer  $\geq$  0)

**C** Component number; required if and only if **NØRM** = **PØINT** and **G** is a geometric grid point (one of the integers 1 to 6)

- Remarks:
1. Real eigenvalue extraction data sets must be selected in the case control deck (**METHØD** = **SID**) to be used by **NASTRAN**.
  2. The units of **F1** and **F2** are cycles per unit time.
  3. The continuation card is required.
  4. A nonzero value of **E** in field 9 also modifies the convergence criteria.
  5. If **NØRM** = **MAX**, components that are not in the analysis set may have values larger than unity.
  6. If **NØRM** = **PØINT**, the selected component must be in the analysis set.

Input Data Card FORCE Static Load

Description: Defines a static load at a grid point by specifying a vector.

Format and Example:

1	2	3	4	5	6	7	8	9	10
<u>FORCE</u>	<u>SID</u>	<u>G</u>	<u>CID</u>	<u>F</u>	<u>N1</u>	<u>N2</u>	<u>N3</u>		
<u>FORCE</u>	2	5	6	2.9	0.0	1.0	0.0		

Field

Contents

- SID Load set identification number (Integer > 0)
- G Grid-point identification number (Integer > 0)
- CID Coordinate system identification number (Integer ≥ 0)
- F Scale factor (real)
- N1,N2,N3 Components of vector measured in coordinate system defined by CID (real;  $N1^2 + N2^2 + N3^2 > 0.0$ )

Remarks: 1. The static load applied to grid point G is given by

$$\vec{f} = F \cdot \vec{N}$$

where  $\vec{N}$  is the vector defined in fields 6, 7, and 8.

- 2. Load sets must be selected in the case control deck (LOAD = SID) to be used by NASTRAN.
- 3. A CID of zero references the basic coordinate system.

Input Data Card GRID Grid Point

Description: Defines the location of a geometric grid point of the structural model, the directions of its displacement, and its permanent single-point constraints.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GRID	ID	CP	X1	X2	X3	CD	PS	X	
GRID	2	3	1.0	2.0	3.0	4	316		

Field Contents

- ID Grid-point identification number ( $0 < \text{Integer} < 999999$ )
- CP Identification number of coordinate system in which the location of the grid points is defined ( $\text{Integer} \geq 0$ )
- X1,X2,X3 Location of the grid point in coordinate system CP (real)
- CD Identification number of coordinate system in which displacements, degrees of freedom, constraints, and solution vectors are defined at the grid point ( $\text{Integer} \geq 0$ )
- PS Permanent single-point constraints associated with grid point; any of the digits 1 to 6 with no embedded blanks ( $\text{Integer} \geq 0$ )

- Remarks:
1. All grid-point identification numbers must be unique with respect to all other structural points.
  2. The meaning of X1, X2, and X3 depends on the type of coordinate system CP as follows: (see the coordinate system bulk data cards)

Type	X1	X2	X3
Rectangular	X	Y	Z
Cylindrical	R	$\theta$ , deg	Z
Spherical	R	$\theta$ , deg	$\phi$ , deg

3. The collection of all CD coordinate systems defined on all GRID cards is called the global coordinate system. All degrees of freedom, constraints, and solution vectors are expressed in the global coordinate system.

Input Data Card LOAD Static Load Combination (Superposition)

Description: Defines a static load as a linear combination of load sets that are defined by using FORCE and MOMENT cards.

Format and Example:

1	2	3	4	5	6	7	8	9	10
<u>LOAD</u>	SID	S	S1	L1	S2	L2	S3	L3	abc
<u>LOAD</u>	101	-0.5	1.0	3	6.2	4			
+bc	S4	L4		-etc.-					

(etc.)

Field

Contents

- SID Load set identification number (Integer > 0)
- S Scale factor (real)
- Si Scale factors (real)
- Li Load set identification numbers defined by means of card types previously enumerated (Integer > 0)

Remarks: 1. The load vector defined is given by

$$\{P\} = S \sum_i S_i \{P_{Li}\}$$

- The Li must be unique. The remainder of the physical card containing the last entry must be blank.
- Load sets must be selected in the case control deck (LOAD = SID) to be used by NASTRAN.
- A LOAD card may not reference a set identification number defined by another LOAD card.

Input Data Card MAT1 Material Property Definition

Description: Defines the material properties for linear, temperature-independent, isotropic materials.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MAT1	MID	E	G	NU	RHØ				+abc
MAT1	17	3.+7	1.9+7		4.28				ABC
+abc	ST	SC	SS						
+BC	20.+4	15.+4	12.+4						

<u>Field</u>	<u>Contents</u>
MID	Material identification number (Integer > 0)
E	Young's modulus (Real $\geq$ 0.0, or blank)
G	Shear modulus (Real $\geq$ 0.0, or blank)
NU	Poisson's ratio (-1.0 < Real $\leq$ 0.5, or blank)
RHØ	Mass density (real)
ST,SC,SS	Stress limits for tension, compression, and shear, respectively (real)

- Remarks:
1. Either E or G must be positive (i.e., either  $E > 0.0$  or  $G > 0.0$  or both E and G may be  $>0.0$ ).
  2. If either E, G, or NU is blank, it will be computed to satisfy the identity  $E = 2(1+NU)G$ ; otherwise, values supplied by the user will be used.
  3. The material identification number must be unique for all MAT1 cards.
  4. If E and NU or G and NU are both blank, they will both be given the value 0.0.

Input Data Card MOMENT Static Moment

Description: Defines a static moment at a grid point by specifying a vector.

Format and Example:

	1	2	3	4	5	6	7	8	9	10
<u>MOMENT</u>	SID	G	CID	M	N1	N2	N3			
<u>MOMENT</u>	2	5	6	2.9	0.0	1.0	0.0			

Field

Contents

- SID Load set identification number (Integer > 0)
- G Grid-point identification number (Integer > 0)
- CID Coordinate system identification number (Integer ≥ 0)
- M Scale factor (real)
- N1,N2,N3 Components of vector measured in coordinate system defined by CID (real;  $N1^2 + N2^2 + N3^2 > 0.0$ )

Remarks: 1. The static moment applied to grid point G is given by

$$\vec{m} = M \cdot (N1, N2, N3)$$

- 2. Load sets must be selected in the case control deck (LOAD = SID) to be used by NASTRAN.
- 3. A CID of zero references the basic coordinate system.

Input Data Card ØMIT Omitted Coordinates

Description: Defines coordinates (degrees of freedom) that the user desires to omit from the problem through matrix partitioning. Used to reduce the number of independent degrees of freedom.

Format and Example:

1	2	3	4	5	6	7	8	9	10
<u>ØMIT</u>	ID	C	ID	C	ID	C	ID	C	
<u>ØMIT</u>	16	2	23	3516			1	4	

Field Contents

- ID            Grid-point identification number (Integer > 0)
- C            Component number; any unique combination of the digits 1 to 6 for grid points

- Remarks:
1. Coordinates specified on ØMIT cards may not be specified on SUPØRT or SPC cards nor as permanent single-point constraints on GRID card.
  2. As many as 24 coordinates may be omitted by a single card.

Input Data Card PARAM Parameter

Description: Specified values for parameters used in DMAP sequences, including rigid formats.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PARAM	N	V1							
PARAM	CØUPMASS								

Field

Contents

N

CØUPMASS

V1

A positive integer value of this parameter causes the generation of coupled mass matrices rather than lumped mass matrices for all bar elements that include bending stiffness. This option applies to both structural and nonstructural mass for BAR and TRIA2. A negative value causes the generation of lumped mass matrices (translational components only) for both BAR and TRIA2. (This is the default.)



Input Data Card PBAR Simple Beam Property

Description: Defines the properties of a simple beam (bar) which is used to create bar elements by means of the CBAR card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PBAR	PID	MID	A	I1	I2	J	NSM	X	abc
PBAR	39	6	2.9		5.97				123
+bc	C1	C2	D1	D2	E1	E2	F1	F2	def
+23			2.0	4.0					
+ef	K1	K2	I12						

<u>Field</u>	<u>Contents</u>
PID	Property identification number (Integer > 0)
MID	Material identification number (Integer > 0)
A	Area of bar cross section (real)
I1, I2, I12	Area moments of inertia (real, $I_1 I_2 \geq I_{12}^2$ )
J	Torsional constant (real)
NSM	Nonstructural mass per unit length (real)
K1, K2	Area factor for shear (real)
Ci, Di, Ei, Fi	Stress recovery coefficients (real)

Remarks: 1. For structural problems, PBAR cards may only reference MAT1 material cards.

Input Data Card PSHEAR Shear Panel Property

Description: Defines the elastic properties of a shear panel. Referenced by the CSHEAR card.

Format and Example:

1	2 3 4 5				6 7 8 9				10
PSHEAR	PID	MID	T	NSM	PID	MID	T	NSM	
PSHEAR	13	2	4.9	16.2	14	6	4.9	14.7	

Field

Contents

PID Property identification number (Integer > 0)  
MID Material identification number (integer > 0)  
T Thickness of shear panel (Real ≠ 0.0)  
NSM Nonstructural mass per unit area (real)

- Remarks:
1. All PSHEAR cards must have unique identification numbers.
  2. PSHEAR cards may only reference MAT1 material cards.
  3. One or two shear panel properties may be defined on a single card.

**Input Data Card PTRIA2 Homogeneous Triangular Element Property**

**Description:** Defines the properties of a homogeneous triangular element of the structural model including membrane, bending, and transverse shear effects. Referenced by the CTRIA2 card.

**Format and Example:**

1	2 3 4 5				6 7 8 9				10
PTRIA2	PID	MID	T	NSM	PID	MID	T	NSM	
PTRIA2	2	16	3.92	14.7	6	16	2.96		

**Field**

**Contents**

**PID** Property identification number (Integer > 0)  
**MID** Material identification number (Integer > 0)  
**T** Thickness (Real > 0.0)  
**NSM** Nonstructural mass per unit area (real)

- Remarks:**
1. All PTRIA2 cards must have unique identification numbers.
  2. The thickness used to compute the membrane and transverse shear properties is T.
  3. The area moment of inertia per unit width that is used to compute the bending stiffness is  $T^3/12$ .
  4. Outer fiber distances of  $\pm T/2$  are assumed.
  5. One or two homogeneous triangular element properties may be defined on a single card.

Input Data Card PTRMEM Triangular Membrane Property

Description: Used to define the properties of a triangular membrane element. Referenced by the CTRMEM card. No bending properties are included.

Format and Example:

1	2 3 4 5				6 7 8 9				10
PTRMEM	PID	MID	T	NSM	PID	MID	T	NSM	
PTRMEM	17	23	4.25	0.2					

<u>Field</u>	<u>Contents</u>
PID	Property identification number (Integer > 0)
MID	Material identification number (Integer > 0)
T	Membrane thickness (Real > 0.0)
NSM	Nonstructural mass per unit area (real)

- Remarks:
1. All PTRMEM cards must have unique property identification numbers.
  2. One or two triangular membrane properties may be defined on a single card.

Input Data Card SPC Single-Point Constraint

Description: Defines sets of single-point constraints and enforced displacements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPC	SID	G	C	D	G	C	D		
SPC	2	32	436	-2.6	5		+2.9		

<u>Field</u>	<u>Contents</u>
SID	Identification number of single-point constraint set (Integer > 0)
G	Grid-point identification number (Integer > 0)
C	Component number; up to six unique such digits may be placed in the field with no embedded blanks ( $6 \geq \text{Integer} \geq 0$ )
D	Value of enforced displacement for all coordinates designated by G and C (real)

- Remarks:
1. A coordinate referenced on this card may not be referenced on an OMIT or SUPORT card. D must be 0.0 for dynamics problem.
  2. Single-point forces of constraint are recovered during stress data recovery.
  3. Single-point constraint sets must be selected in the case control deck (SPC = SID) to be used by NASTRAN.
  4. One to twelve single-point constraints may be defined on a single card.
  5. SPC degrees of freedom may be redundantly specified as permanent constraints on the GRID card.

Input Data Card SPCADD Single-Point Constraint

Description: Defines a single-point constraint set as a union of single-point constraint sets that are defined by means of SPC cards.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPCADD	SID	S1	S2	S3	S4	S5	S6	S7	abc
SPCADD	101	3	2	9	1				
+bc	S8	S9	-etc.-						

-etc.-

Field

Contents

- SID Identification number for new single-point constraint set  
(Integer > 0)
- Si Identification numbers of single-point constraint sets defined  
by means of SPC cards (Integer > 0; SID ≠ Si)

- Remarks:
1. Single-point constraint sets must be selected in the case control deck (SPC = SID) to be used by NASTRAN.
  2. No Si may be the identification number of a single-point constraint set defined by another SPCADD card.
  3. The Si values must be unique.

Input Data Card SUPØRT Fictitious Support

Description: Defines coordinates at which the user desires determinate reactions to be applied to a free body during analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SUPØRT	ID	C	ID	C	ID	C	ID	C	
SUPØRT	16	215							

<u>Field</u>	<u>Contents</u>
ID	Grid-point identification number (Integer > 0)
C	Component number; any unique combination of the digits 1 to 6 for grid points

- Remarks:
1. Coordinates defined on this card may not appear on single-point cards (SPC) or omit cards (ØMIT).
  2. One to twenty-four support coordinates may be defined on a single card.

Input Data Card \$ Comment

Description: For the user's convenience in inserting commentary material into the unsorted echo of the input bulk data deck. The \$ card is otherwise ignored by the program. These cards will not appear in a sorted echo.

Format and Example:

1	2	3	4	5	6	7	8	9	10	
\$	followed by any legitimate characters in card columns 2 to 80									
\$	THIS IS A REMARK (*,'\$\$)--/									



### 2.3 Case Control Deck

The case control cards that are used for selecting items from the bulk data deck, output selecting, and subcase definition are listed below. The first four characters of the mnemonic are sufficient.

DISPLACEMENT	requests the displacements for a selected set of grid points
ECHØ	selects echo options for the bulk data deck; default is a sorted bulk data echo
ELFØRCE	requests the forces in a set of structural elements
LABEL	defines a text to be printed on the third line of each page of output
LINE	sets the number of data lines per printed page; default is 50
LØAD	selects static loading condition
MAXLINES	sets the maximum number of output lines; default is 20,000
METHØD	selects the conditions for eigenvalue analysis
ØLØAD	selects a set of applied loads for output
ØUTPUT	delimits the various output packets
PLØTID	plotter identification
SDISPLACEMENT	requests the displacements of the independent components for a selected set of points or modal coordinates
SET	defines lists of point numbers, element numbers, or frequencies for use in output requests
SPC	selects a set of single-point constraints
SPCFØRCES	requests the single-point forces of a constraint at a set of points
STRESS	requests the stresses in a set of structural elements

**SUBCASE** defines the beginning of a subcase that is terminated by the next subcase delimiters encountered

**SUBTITLE** defines a text to be printed on the second line of each page of output

**TITLE** defines a text to be printed on the first line of each page of output

**\$** comment card

Case Control Data Card DISPLACEMENT Displacement Output Request

Description: Requests the form and type of displacement vector output.

Format and Example(s):

$$\text{DISPLACEMENT} \left[ \begin{array}{l} (\text{SØRT1}, \text{PRINT}) \\ (\text{SØRT2}, \text{PUNCH}) \end{array} \right] = \left\{ \begin{array}{l} \text{ALL} \\ n \\ \text{NONE} \end{array} \right\}$$

DISPLACEMENT = 5

DISPLACEMENT(REAL) = ALL

DISPLACEMENT(SØRT2, PUNCH) = ALL

<u>Option</u>	<u>Meaning</u>
SØRT1	Output will be presented as a tabular listing of grid points for each load, frequency, or eigenvalue.
SØRT2	Output will be presented as a tabular listing of load or frequency for each grid point. SØRT2 is available only in static analysis.
PRINT	The printer will be the output media.
PUNCH	The card punch will be the output media.
ALL	Displacements for all points will be output.
NONE	Displacements for no points will be output.
n	Set identification of previously appearing SET card. Only the displacements of points whose identification numbers appear on this SET card will be output (Integer > 0).

- Remarks:
1. Both PRINT and PUNCH may be used.
  2. In static analysis problems, any request for SØRT2 causes all output to be SØRT2.
  3. DISPLACEMENT = NONE allows overriding of an overall output request.

Case Control Data Card ECHØ Bulk Data Echo Request

Description: Requests the echo of bulk data deck.

Format and Example(s):

ECHØ = 

<u>SØRT</u>
UNSØRT
BØTH
NØNE
PUNCH

ECHØ = BØTH

ECHØ = PUNCH, SØRT

<u>Option</u>	<u>Meaning</u>
SØRT	Sorted echo will be printed.
UNSØRT	Unsorted echo will be printed.
BØTH	Both sorted and unsorted echo will be printed.
NØNE	No echo will be printed.
PUNCH	The sorted bulk data deck will be punched onto cards.

- REMARKS:
1. If no ECHØ card appears, a sorted echo will be printed.
  2. Unrecognizable options will be treated as SØRT.
  3. Any option overrides the default. Thus, for example, if both print and punch are desired, both SØRT and PUNCH must be requested on the same card.

Case Control Data Card ELFØRCE Element Force Output Request

Description: Requests the form and type of element force output.

Format and Example(s):

$$\text{ELFØRCE} \left[ \left( \frac{\text{SØRT1}}{\text{SØRT2}}, \frac{\text{PRINT}}{\text{PUNCH}} \right) \right] = \left\{ \begin{array}{c} \text{ALL} \\ n \\ \text{NØNE} \end{array} \right\}$$

ELFØRCE = ALL

ELFØRCE(PUNCH, PRINT) = 17

ELFØRCE = 25

<u>Option</u>	<u>Meaning</u>
SØRT1	Output will be presented as a tabular listing of elements for each load, frequency, or eigenvalue.
SØRT2	Output will be presented as a tabular listing of load or frequency for each element type. SØRT2 is available only in static analysis.
PRINT	The printer will be the output media.
PUNCH	The card punch will be the output media.
ALL	Forces for all elements will be output.
NØNE	Forces for no elements will be output.
n	Set identification of a previously appearing SET card. Only forces of elements whose identification numbers appear on this SET card will be output (Integer > 0).

- Remarks:
1. Both PRINT and PUNCH may be requested.
  2. In static analysis problems, any request for SØRT2 output causes all output to be SØRT2.
  3. ELFØRCE = NØNE allows overriding of an overall request.

Case Control Data Card LABEL Output Label

Description: Defines a BCD label which will appear on the third heading line of each page of the NASTRAN printer output.

Format and Example(s):

LABEL = {Any BCD data}

LABEL = PROBLEM

- Remarks:
1. LABEL appearing at the subcase level will label output for that subcase only.
  2. LABEL appearing before all subcases will label any outputs which are not subcase dependent.
  3. If no LABEL card is supplied, the label line will be blank.
  4. LABEL information is also placed on the NASTRAN plotter output when applicable.

Case Control Data Card LINE Data Lines Per Page

Description: Defines the number of data lines per printed page.

Format and Example(s):

$$\text{LINE} = \left\{ \begin{array}{c} 50 \\ - \\ n \end{array} \right\} \text{ IBM or CDC}$$

$$\text{LINE} = \left\{ \begin{array}{c} 45 \\ - \\ n \end{array} \right\} \text{ UNIVAC}$$

LINE = 35

Option

Meaning

n                      Number of data lines per page (Integer > 0)

- Remarks:
1. If no LINE card appears, the appropriate default is used.
  2. For 11-inch paper, 50 is the recommended number; for  $8\frac{1}{2}$ -inch paper, 35 is the recommended number.

Case Control Data Card LØAD External Static Load Set Selection

Description: Selects the external static load set to be applied to the structural model.

Format and Example(s):

LØAD = n

LØAD = 15

Option

Meaning

n Set identification of at least one external load card and hence must appear on at least one FORCE, MOMENT, or LØAD card (Integer > 0).

- Remarks:
1. The above static load cards will not be used by NASTRAN unless selected in case control.
  2. LØAD is only applicable in statics and buckling problems.
  3. The total load applied will be the sum of external (LØAD) and constrained displacement (SPC) loads.
  4. Static loads should have unique set identification numbers.



Case Control Data Card MAXLINES Maximum Number of Output Lines

Description: Sets the maximum number of output lines to a given value.

Format and Example(s):

$$\text{MAXLINES} = \left\{ \frac{20000}{n} \right\}$$

MAXLINES = 50000

Option

Meaning

n Maximum number of output lines which the user wishes to allow  
(Integer > 0)

- Remarks:
1. Any time this number is exceeded, NASTRAN will terminate through PEXIT.
  2. This card may or may not override the system operating control cards. Users should check with the local operations staff.

Case Control Data Card METHØD Real Eigenvalue Extraction Method Selection

Description: Selects the real eigenvalue parameters to be used by the READ module.

Format and Example(s):

METHØD = n

METHØD = 33

<u>Option</u>	<u>Meaning</u>
n	Set identification number of an EIGR card (normal modes or modal formulation) or an EIGB card (buckling) (Integer > 0).

Remarks: 1. An eigenvalue extraction method must be selected when extracting real eigenvalues by using the functional module READ.

Case Control Data Card ØLØAD Applied Load Output Request

Description: Requests the form and type of applied load vector output.

Format and Example(s):

$$\text{ØLØAD} \left[ \left( \frac{\text{SØRT1}}{\text{SØRT2}}, \frac{\text{PRINT}}{\text{PUNCH}} \right) \right] = \left\{ \begin{array}{c} \text{ALL} \\ n \\ \text{NONE} \end{array} \right\}$$

ØLØAD = ALL

ØLØAD(SØRT1) = 5

<u>Option</u>	<u>Meaning</u>
SØRT1	Output will be presented as a tabular listing of grid points for each load, frequency, or eigenvalue.
SØRT2	Output will be presented as a tabular listing of load or frequency for each grid point. SØRT2 is available only in static analysis.
PRINT	The printer will be the output media.
PUNCH	The card punch will be the output media.
ALL	Applied loads for all points will be output. (SØRT1 will only output nonzero values.)
NØNE	Applied loads for no points will be output.
n	Set identification of previously appearing SET card. Only loads on points whose identification numbers appear on this SET card will be output (Integer > 0).

- Remarks:
1. Both PRINT and PUNCH may be requested.
  2. In static analysis problems, any request for SØRT2 output causes all output to be SØRT2.
  3. A request for SØRT2 causes loads (zero and nonzero) to be output.
  4. ØLØAD = NØNE allows overriding of an overall output request.

Case Control Data Card ØUTPUT Output Packet Delimiter

Description: Delimits the various output packets, structure plotter, and printer/punch.

Format and Example(s):

ØUTPUT [(PLØT)]

ØUTPUT

ØUTPUT(PLØT)

Option

Meaning

No qualifier Beginning of printer output packet; this is not a required card.

PLØT Beginning of structure plotter packet. This card must precede all structure plotter control cards.

Remarks: 1. The structure plotter packet must be at the end of the case control deck.

2. The delimiting of a printer packet is completely optional.

Case Control Data Card PLØTID Plotter Identification

Description: Defines the BCD identification which will appear on the first frame of any NASTRAN plotter output.

Format and Example(s):

PLØTID = {Any BCD data}

PLØTID = TEST PLØT

- Remarks:
1. PLØTID must appear before the ØUTPUT(PLØT) card.
  2. The presence of PLØTID causes a special header frame to be plotted, with the supplied identification plotted several times. This allows easy identification of NASTRAN plotter output.
  3. If no PLØTID card appears, no ID frame will be plotted.
  4. The PLØTID header frame will not be generated for the table plotters.

Description: Requests the form and type of solution set displacement output.

Format and Example(s):

$$\text{SDISPLACEMENT} \left[ \left( \frac{\text{SORT1}}{\text{SORT2}}, \frac{\text{PRINT}}{\text{PUNCH}} \right) \right] = \left\{ \begin{array}{c} \text{ALL} \\ n \\ \text{NONE} \end{array} \right\}$$

SDISPLACEMENT = ALL

SDISPLACEMENT(SORT2, PUNCH) = NONE

<u>Option</u>	<u>Meaning</u>
SORT1	Output will be presented as a tabular listing of grid points for each load, frequency, or eigenvalue.
SORT2	Output will be presented as a tabular listing of frequency for each grid point (or mode number).
PRINT	The printer will be the output media.
PUNCH	The card punch will be the output media.
ALL	Displacements for all points (modes) will be output.
NONE	Displacements for no points (modes) will be output.
n	Set identification of previously appearing SET card. Only displacements of points whose identification numbers appear on this SET card will be output (Integer > 0).

Remarks: 1. Both PRINT and PUNCH may be requested.

2. SDISPLACEMENT = NONE allows overriding of an overall output request.

Case Control Data Card SET Set Definition Card

Description: Lists identification numbers (point or element) for output requests.

Format and Example(s):

(1) SET n = {i<sub>1</sub> [, i<sub>2</sub>, i<sub>3</sub> THRU i<sub>4</sub> EXCEPT i<sub>5</sub>, i<sub>6</sub>, i<sub>7</sub>, i<sub>8</sub> THRU i<sub>9</sub>]}

SET 77 = 5

SET 88 = 5, 6, 7, 8, 9, 10 THRU 55 EXCEPT 15, 16, 77, 78, 79, 100 THRU 300

SET 99 = 1 THRU 100000

(2) SET n = {r<sub>1</sub> [, r<sub>2</sub>, r<sub>3</sub>, r<sub>4</sub>]}

SET 101 = 1.0, 2.0, 3.0

SET 105 = 1.009, 10.2, 13.4, 14.0, 15.0

<u>Option</u>	<u>Meaning</u>
n	Set identification (Integer > 0). Any set may be redefined by reassigning its identification number. Sets inside SUBCASE delimiters are local to the SUBCASE.
i <sub>1</sub> , i <sub>2</sub> , etc.	Element or point identification number at which output is requested (Integer > 0). If no such identification number exists, the request is ignored.
i <sub>3</sub> THRU i <sub>4</sub>	Output at set identification numbers i <sub>3</sub> to i <sub>4</sub> (i <sub>4</sub> > i <sub>3</sub> ).
EXCEPT	Set identification numbers following EXCEPT will be deleted from output list as long as they are in the range of the set defined by the immediately preceding THRU.
r <sub>1</sub> , r <sub>2</sub> , etc.	Frequencies for output (Real > 0.0). The nearest solution frequency will be output. EXCEPT and THRU cannot be used.

- Remarks:
1. A SET card may be more than one physical card. A comma at the end of a physical card signifies a continuation card. Commas may not end a set.
  2. Set identification numbers following EXCEPT that are within the range of the THRU must be in ascending order.

Case Control Data Card SPC Single-Point Constraint Set Selection

Description: Selects the single-point constraint set to be applied to the structural model.

Format and Example(s):

SPC = n

SPC = 10

Option

Meaning

n Set identification of a single-point constraint set and hence must appear on an SPC or SPCADD card (Integer > 0).

Remarks: 1. SPC or SPCADD cards will not be used by NASTRAN unless selected in case control.



Description: Requests the form and type of single-point force of the constraint vector output.

Format and Example(s):

$$\text{SPCFORCES} \left[ \left( \frac{\text{SORT1}}{\text{SORT2}}, \frac{\text{PRINT}}{\text{PUNCH}} \right) \right] = \left\{ \begin{array}{c} \text{ALL} \\ n \\ \text{NONE} \end{array} \right\}$$

SPCFORCES = 5

SPCFORCES(SORT2, PUNCH, PRINT) = ALL

SPCFORCES(PHASE) = NONE

<u>Option</u>	<u>Meaning</u>
SORT1	Output will be presented as a tabular listing of grid points for each load, frequency, or eigenvalue.
SORT2	Output will be presented as a tabular listing of load or frequency for each grid point.
PRINT	The printer will be the output media.
PUNCH	The card punch will be the output media.
ALL	Single-point forces of constraint for all points will be output. (SORT1 will only output nonzero values.)
NONE	Single-point forces of constraint for no points will be output.
n	Set identification of previously appearing SET card. Only single-point forces of constraint for points whose identification numbers appear on this SET card will be output (Integer > 0).

- Remarks:
- Both PRINT and PUNCH may be requested.
  - In static analysis, any request for SORT2 output causes all output to be SORT2.
  - A request for SORT2 causes loads (zero and nonzero) to be output.
  - SPCFORCES = NONE allows overriding of an overall output request.

Case Control Data Card STRESS Element Stress Output Request

Description: Requests the form and type of element stress output.

Format and Example(s):

$$\text{STRESS} \left[ \left( \frac{\text{SØRT1}}{\text{SØRT2}}, \frac{\text{PRINT}}{\text{PUNCH}} \right) \right] = \left\{ \begin{array}{c} \text{ALL} \\ n \\ \text{NØNE} \end{array} \right\}$$

STRESS = 5

STRESS = ALL

STRESS (SØRT1, PRINT, PUNCH) = 15

<u>Option</u>	<u>Meaning</u>
SØRT1	Output will be presented as a tabular listing of elements for each load, frequency, or eigenvalue.
SØRT2	Output will be presented as a tabular listing of load or frequency for each element type. SØRT2 is available only in static analysis.
PRINT	The printer will be the output media.
PUNCH	The card punch will be the output media.
ALL	Stresses for all elements will be output.
NØNE	Stresses for no points will be output.
n	Set identification of a previously appearing SET card (Integer > 0). Only stresses for elements whose identification numbers appear on this SET card will be output.

- Remarks:
1. Both PRINT and PUNCH may be requested.
  2. In static analysis, any request for SØRT2 output causes all output to be SØRT2.
  3. STRESS = NØNE allows overriding of an overall output request.

Case Control Data Card SUBCASE Subcase Delimiter

Description: Delimits and identifies a subcase.

Format and Example(s):

SUBCASE n

SUBCASE 101

Option

Meaning

n Subcase identification number (Integer > 0).

Remarks: 1. The subcase identification number n must be strictly increasing (i.e., greater than all previous subcase identification numbers).

2. Plot requests refer to n.

Case Control Data Card SUBTITLE Output Subtitle

Description: Defines a BCD subtitle which will appear on the second heading line of each page of the NASTRAN printer output.

Format and Example(s):

SUBTITLE = {Any BCD data}

SUBTITLE = NASTRAN PROBLEM NO. 5-1A

- Remarks:
1. SUBTITLE appearing on the subcase level will title output for that subcase only.
  2. SUBTITLE appearing before all subcases will title any outputs which are not subcase dependent.
  3. If no SUBTITLE card is supplied, the subtitle line will be blank.
  4. SUBTITLE information is also placed on the NASTRAN plotter output as applicable.

Case Control Data Card TITLE Output Title

Description: Defines a BCD title which will appear on the first heading line of each page of the NASTRAN printer output.

Format and Example(s):

TITLE = {Any BCD data}

TITLE = \*\*\$// ABCDEFGHI .... \$

- Remarks:
1. TITLE appearing at the subcase level will title output for that subcase only.
  2. TITLE appearing before all subcases will title any outputs which are not subcase dependent. •
  3. If no TITLE card is supplied, the title line will contain data and page numbers only.
  4. TITLE information is also placed on NASTRAN plotter output as applicable.

Case Control Data Card \$ Comment Card

Description: Defines a comment card by specifying a \$ in column 1 with commentary text appearing in columns 2 to 80.

Format and Example(s):

\$ {Any BCD data}

\$---THIS IS AN EXAMPLE OF A COMMENT CARD.

Remarks: 1. Unlike other case control cards which are free field, the comment card must have the \$ in column 1.

## 2.4 Executive Control Deck

The executive control deck begins with the NASTRAN ID card and ends with the CEND card. It identifies the job and declares the general conditions under which the job is to be executed. The completed DMAP sequence must appear in the executive control deck.

The format of the executive control cards is free field. The name of the operation (e.g., TIME) is separated from the operand by one or more blanks. The fields in the operand are separated by commas and may be up to eight integers (K) or alphanumeric (A<sub>i</sub>), as indicated in the following control card descriptions. The first character of an alphanumeric field must be alphabetic, followed by up to seven additional alphanumeric characters. Blank characters may be placed adjacent to separating commas if desired.

The following control cards can be used in the executive control deck:

### ID A1, A2 (required)

A1, A2 - Any legal alphanumeric fields chosen by the user for problem identification.

### APP DMAP (required)

DMAP sequence is to follow, must begin with BEGIN \$ or XDMAP ... \$ card and end with END \$ card.

### TIME K

K - Maximum allowable execution time in minutes.

### CEND (required)

Indicates end of executive control cards.

## 2.5 NASTRAN Card

The NASTRAN card is used to change the default values for certain operational parameters, such as buffer size and computer model number. The NASTRAN card is optional, but if present must be the first card of the NASTRAN data deck. The NASTRAN card is a free-field card (similar to cards in the executive control deck). Its format is as follows:

NASTRAN keyword<sub>1</sub> = value, keyword<sub>2</sub> = value, . . .

The most frequently used keywords are as follows:

BUFSIZE defines the number of words in a GINØ buffer. Usually this value is standardized at any particular installation. However, the desired value may be different from the default value of 1803 (IBM), 1183 (Control Data), and 871 (UNIVAC).

**CONFIG** defines the model number of the configuration for use in timing equations for matrix operations. Entries exist for the following configurations:

Computer	Configuration	Model number
IBM 360/370	0 (default)	91, 95
	3	50
	4	65
	5	75
	6	85
	7	195
	9	155
	Control Data 6000	0 (default)
6		6400
UNIVAC 1100	0 (default)	1108

The machine type is automatically determined by NASTRAN. If the model number is the default, the **CONFIG** keyword is not needed on the NASTRAN card. It is important to indicate the proper configuration; otherwise, all time-dependent matrix decisions will be incorrect.

**KON360** defines the number of 32-bit words to release for IBM 360 OS routines and FORTRAN buffers. The default is 4096.

**MDCOM(i)** defines a nine-word array for module communications. Currently, only **MDCOM(1)** is supported. When **MDCOM(1) = 999999**, optimization of passive columns in the symmetric decomposition routine is not used. If **MDCOM(1) = 1**, diagnostic statistics from subroutine **SDCOMP** are printed.

**HICORE** defines the amount of open core available to the user on the UNIVAC 1100 series machines. The user area default is nominally 65000 decimal words. The ability to increase this value may be installation limited.

**IPREC** is the precision flag. Default is 1 on Control Data and 2 on IBM and UNIVAC.

### 3. PLOTTING

#### 3.1 General Description of Plotting Capability

NASTRAN provides the capability for generating on any of several different plotters the following kinds of plots:

Undeformed geometric projections of the structural model



Static deformations of the structural model by either displaying the deformed shape (alone or superimposed on the undeformed shape) or displaying the displacement vectors at the grid points (superimposed on either the deformed or undeformed shape)

Modal deformations (sometimes called mode shapes or eigenvectors) that result from real eigenvalue analysis by the same options stated in the previous statement

Requests for structure plots are accomplished in the case control deck by submitting a structure plot request packet. The discussion of these packets constitutes most of the remainder of this section. The optional PLPLOT card is considered part of the plot packets although it must precede any OUTPUT(PLPLOT) card.

To create plots, a plotter and model name must be specified by the user. The method used to specify this information may vary according to the plot request made, but the actual names used do not vary. In addition, a physical plot tape must be set up by the user. The control cards needed to set up a plot tape are generally installation dependent. There is one plot tape, PLT2.

The following table is a list of permissible plotter and model names, together with the corresponding plot tapes which must be set up by the user. The underlined items are the default models for each plotter. A model name is generally specified as two items, each having a default value. The default value of the second item is in some cases dependent upon the value specified for the first item. If no plotter is specified by the user, the requested plots will be created for the Stromberg-Carlson (SC) 4020 microfilm plotter.

Plotter name	Plot tape	Plotter model
<u>CalComp</u>	<u>PLT2</u>	$\left\{ \begin{array}{c} \underline{765} \\ 763 \end{array} \right\}, \left\{ \begin{array}{c} \underline{205} \\ 210 \\ 105 \\ 110 \end{array} \right\}$ $\left\{ \begin{array}{c} \underline{565} \\ 563 \end{array} \right\}, \left\{ \begin{array}{c} \underline{205} \\ 210 \\ 105 \\ 110 \\ 305 \\ 310 \end{array} \right\}$
<u>NASTPLT</u>	<u>PLT2</u>	$\left\{ \begin{array}{c} \underline{M} \\ T \\ D \end{array} \right\}, \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\}$
<u>SC</u>	<u>PLT2</u>	<u>4020</u>

The plotter name CalComp is used for California Computer Products, Inc. plotters. The default model is a 765, 205. The first model item is the plotter model number as used in CalComp hardware descriptions. The 700 series plotters are those having the ZIP mode and 24 incremental directions. The 500 series plotters are those having only 8 incremental directions. The 600 series may have either 24 or 8 incremental directions. If the user has access to only a 663 or 665 plotter, it should be specified as a 563 or 565 if it has only 8 incremental directions, and as a 763 or 765 if it has 24 incremental directions. The 563 and 763 are both 30-in. drum plotters, and the 565 and 765 are both 12-in. drum plotters.

The second model item indicates the type of tape transport used with the CalComp plotter and the increment size of the plotter. There are two possible increment sizes, .010 and .005 in. The last two digits of this second model item represent these two possible increment sizes (i.e., 10 = .010 and 05 = .005). The first digit of the second model item represents the type of tape transport attached to the plotter. There are three types of tape transports available. The primary differences among these transports are the number of characters needed to cause one incremental movement on the plotter. Some transports (e.g., the 470, 570, and 750 models) require three characters. These transports can only be attached to the 500 series plotters. Other transports (e.g., the 760 and 770) require two characters for each incremental movement. Still other transports (e.g., the 780) require only one character for each incremental movement. The first digit of the second model item is the number of characters required by the tape transport for each incremental movement. An example of a legitimate CalComp model name is (763,105). This represents a 763 having a 30-in. drum plotter with an increment size of .005 in. and driven by a tape transport requiring only one character for each incremental movement (e.g., a 780 tape transport).

The plotter name NASTPLT is used for the NASTRAN general purpose plotter package. This plotter package is used if the desired plotter is not available in the NASTRAN plotting software. However, if this package is specified, a separate program must be written to interpret the resulting plot tape and to create the corresponding plots on the actual plotter desired. The default model is M,0. The first model item may either be M, T, or D, which indicates whether the actual plotter is a microfilm plotter, table plotter, or drum plotter, respectively. The second model item indicates whether or not the actual plotter has any typing capability (i.e., 0 means typing possible, 1 means no typing possible). If no typing capability exists, all printed characters will be drawn. The default plotter type is a microfilm plotter with typing capability. An example of an acceptable model is (T,1). This represents a table plotter having no typing capability. See section 3.3 for more detailed information.

The plotter name SC is used for Stromberg-Carlson Corporation plotters. The only permissible model is the 4020 microfilm plotter. If the only available plotter model is a 4060, the user should determine if it has a 4020 compatibility package, as is usually the case, so as to avoid using the NASTRAN general purpose plotter.

### 3.2 Structure Plotting

In order to assist NASTRAN users in both the preparation of the analytical model and the interpretation of output, the structure plotter provides the following capabilities for undeformed structures:

Placing a symbol at the grid-point locations (optional)

Identifying the grid points by placing the grid-point identification number to the right of the grid-point locations (optional)

Identifying elements by placing the element identification number and element symbol at the center of each element (optional)

Connecting the grid points in a predetermined manner by using the structural elements

The following capabilities are provided for deformed structures:

Placing a symbol at the deflected grid-point location (optional)

Identifying the deflected grid points by placing the grid-point identification number to the right of the deflected grid-point locations (optional)

Connecting the deflected grid points in a predetermined manner by using the structural elements

Drawing lines that originate at the undeflected or deflected grid-point location to user-specified scale in order to represent the X,Y,Z components or resultant summations of the grid-point deflections

The preceding plots are available in either orthographic, perspective, or stereoscopic projections on several plotters. Stereoscopic plots are normally made only on microfilm plotters since a stereoscopic viewer or projector must be used to obtain the stereoscopic effect. A request for structure plotting is made in the case control deck by means of a plot request packet, which includes all cards from an OUTPUT(PL~~O~~T) card to a BEGIN BULK card. It should be noted that only elements can be plotted. Grid points that are not associated with elements cannot be plotted.

The data card format is free field and subject to rules in the following paragraphs. The cards are basically sequence dependent even though some interchanging in sequence of defining parameters is permissible. The elements and grid points to be plotted may be defined anywhere in the submittal, but the parameters describing the characteristics of the plot are evaluated on the current basis every time a PL~~O~~T or FIND card is encountered. To minimize mistakes, it is suggested that a strict sequence dependency be assumed.

### 3.2.1 Rules for Free-Field Card Specification

1. Only columns 1 to 72 are available. Any information specified in columns 73 to 80 is ignored.
2. If the last character on a card is a comma (not necessarily in column 72), the next card is a continuation of this physical card. Any number of continuation cards may be specified, and together they form a logical card.
3. The mnemonics or values can be placed anywhere on the card but must be separated by delimiters. The following delimiters are used:
  - a. Blank
  - b. Comma
  - c. Left parenthesis
  - d. Right parenthesis
  - e. Equal sign

All these delimiters can be used as needed to aid the legibility of the data.

### 3.2.2 Plot Request Packet Card Format

In the plot request packet card descriptions, the following notations will be used to describe the card format:

Upper-case letters must be punched exactly as shown.

Lower-case letters indicate that a substitution must be made.

Braces { } indicate that a choice of the contents is mandatory.

Brackets [ ] contain an option that may be omitted or included by the user.

Underlined options or values are those for which a default option or an initialized (or computed) value was programmed.

A physical card consists of information punched in columns 1 to 72 of a card.

A logical card may consist of more than one physical card through the use of continuation cards.

Numerical values may always be either integer or real numbers even though a specific type is at times suggested in order to conform to the input in other sections of the program.

### 3.2.3 Plot Titles

Up to four lines of title information will be printed in the lower left-hand corner of each plot. The text for the top three lines is taken from the TITLE, SUBTITLE, and LABEL cards in the case control deck. (See section 2.2 for a description of the TITLE, SUBTITLE, and LABEL cards.) The text for the bottom line may be of two forms, depending on the type plot requested. One form contains the word UNDEFORMED SHAPE. The other form contains the type of plot (statics, modal, etc.), the subcase number, the load set or mode number, and the frequency or eigenvalue or time.

The sequence number for each plot is printed in the upper corners of each frame. The sequence number is determined by the relative position of each PLOT execution card in the plot package. The date and, for deformed plots, the maximum deformation are also printed at the top of each frame.

### 3.2.4 Plot Request Packet Card Descriptions

The general form for each card of the plot request packet is shown enclosed in a rectangular box. A description of the card contents then follows for each card.

### 3.2.5 SET Definition Cards

These cards specify sets of elements that correspond to portions of the structure; they may be referenced by PLOT and FIND cards. The SET card is required.

Each set of elements defines by implication a set of grid points connected by those elements. The set may be modified by deleting some of its grid points. The elements are used for creating the plot itself and labeling the element; the grid points are used for labeling, symbol printing, and drawing deformation vectors.

SET i [INCLUDE] [ELEMENTS] j <sub>1</sub> , j <sub>2</sub> , j <sub>3</sub> THRU j <sub>4</sub> , j <sub>5</sub> , etc. [INCLUDE EXCLUDE EXCEPT] [ELEMENTS GRID POINTS] k <sub>1</sub> , k <sub>2</sub> , k <sub>3</sub> THRU k <sub>4</sub> , k <sub>5</sub> , etc.
--

- i set identification number (positive integer, unique for each set)
- j element identification numbers or element types
- k element identification numbers or grid-point identification numbers or element types

Permissible element types are BAR, TRIA2, TRMEM, and SHEAR.

ALL may be used to select all permissible element types.

INCLUDE may be used at any time for element information. When used with grid points, INCLUDE can be used only to restore previously EXCLUDED grid points. It cannot be used to include grid points in the original set of grid points.

EXCLUDE can be used to delete elements or element types. All grid points that are associated with deleted elements are also deleted. EXCLUDE can be used to delete information vectors from grid points enumerated after an EXCLUDE command.

EXCEPT is a modifier to an INCLUDE or an EXCLUDE statement.

THRU is used to indicate all of the integers in a sequence of identification numbers starting with the integer preceding THRU and ending with the integer following THRU. The integers in the range of the THRU statement need not be consecutive (e.g., the sequence 2, 4, 7, 9 may be specified as 2 THRU 9). THRU is not applicable if element types are specified.

Each SET must be a logical card. Redefinition of sets previously defined is not permitted; however, there is no restriction on the number of sets. The sets of identification numbers can be assembled by use of the word ALL, or by individually listing the integers in any order, such as 1065, 32, 46, 47, 7020 or by listing sequencs using THRU, EXCLUDE, and EXCEPT such as 100 THRU 1000 EXCEPT 182 EXCLUDE 877 THRU 911. Examples of SET cards are as follows:

```
SET 1 INCLUDE 1, 5, 10 THRU 15 EXCEPT 12
```

(Set will consist of elements 1, 5, 10, 11, 13, 14, and 15.)

```
SET 25 = BAR, TRMEM EXCEPT 21
```

(Set will consist of all BAR and TRMEM elements except element 21.)

SET 10 BAR EXCLUDE GRID POINTS 20, 30 THRU 60, EXCEPT 35, 36 INCLUDE  
ELEMENTS 70 THRU 80

(This set will include all bar elements plus elements 70 to 80, and the associated grid-point set will contain all grid points connected by these elements. Grid points 20, 30 to 34, and 37 to 60 will appear on all plots with their symbols and labels; however, no deformation vectors will appear at these grid points when VECTOR is commanded.)

SET (15) = (15 THRU 100) EXCEPT (21 THRU 25)

(This set will include all elements from 15 to 20 and from 26 to 100.)

SET 2 = ALL EXCEPT BAR

(This set will include all elements except bars.)

The equal signs, commas, and parentheses that are used in the preceding examples are delimiters and are not required because blanks also serve as delimiters.

### 3.2.6 Cards Defining Parameters

These cards specify how the structure will be plotted (i.e., type of projection, view angles, scales, etc.). All the multiple-choice parameters are defaulted to a preselected choice if not specified. Each parameter requiring a numerical value that is not specified by the user can either be established internally in the program by means of the FIND card or can assume default values. The FIND card is used to request that the program select a SCALE, ORIGIN, and/or VANTAGE POINT to allow the construction of a plot in a user-specified region of the paper or film.

The parameter cards are listed here in a logical sequence; however, they need not be so specified. Any order may be used, but if a parameter is specified more than once, the value or choice stated last will be used. Each parameter may be either an individual card or a combination of cards on one logical card.

All the parameters used in the generation of the various plots will be printed out as part of the output whether they are directly specified, defaulted, or established using the FIND card.

Initialization of parameters to default values occurs only once. Subsequently, these values remain until altered by a direct input. The only exceptions are the view angles, scale factors, vantage-point parameters, and the origins. Whenever the plotter or the method of projection is changed, the view angles are reset to the default values unless they are respecified by the user. In addition, the scale factors, vantage-point parameters, and the origin must be redefined by the user.

PLOTTER plotter name, MODEL name <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>DENSITY</td> <td> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>800</td></tr> <tr><td>556</td></tr> <tr><td>200</td></tr> </table> </td> <td>BPI</td> </tr> </table>	DENSITY	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>800</td></tr> <tr><td>556</td></tr> <tr><td>200</td></tr> </table>	800	556	200	BPI
DENSITY	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>800</td></tr> <tr><td>556</td></tr> <tr><td>200</td></tr> </table>	800	556	200	BPI	
800						
556						
200						

The plotter names and MODEL names have been listed previously. The tape density information is used only in printout and does not control the density of the generated plot tape. To specify the tape density, use the customary means of communication established at a given installation between the user and the computer operators. This card is required for plotters other than the SC 4020.

<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>ORTHOGRAPHIC</td></tr> <tr><td>PERSPECTIVE</td></tr> <tr><td>STEREOSCOPIC</td></tr> </table>	ORTHOGRAPHIC	PERSPECTIVE	STEREOSCOPIC	PROJECTION
ORTHOGRAPHIC				
PERSPECTIVE				
STEREOSCOPIC				

The default option is orthographic projection. This card is optional.

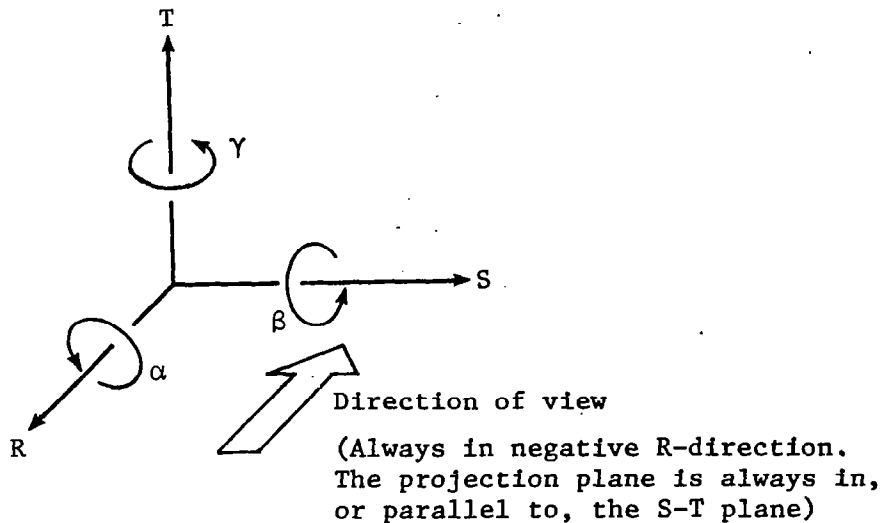
AXES r, s, t <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>SYMMETRIC</td></tr> <tr><td>ANTISYMMETRIC</td></tr> </table>	SYMMETRIC	ANTISYMMETRIC
SYMMETRIC		
ANTISYMMETRIC		
VIEW $\alpha$ , $\beta$ , $\alpha$		

r, s, t X or MX, Y or MY, Z or MZ (where M implies the negative axis)

$\gamma$ ,  $\beta$ ,  $\alpha$  three angles of rotation in degrees (real numbers)

These two parameter cards define the orientation of the object in relation to the observer (i.e., the angles of view). Both of these cards are optional. Defining the observer's coordinate system as R, S, T and the basic coordinates system of the object as X, Y, Z, the angular relationship between the two systems is defined by the three angles  $\gamma$ ,  $\beta$ , and  $\alpha$  as follows:





Using the above convention,  $\gamma$  and  $\beta$  represent the angles of turn and tilt. The default values are:

$$\gamma = 34.27^\circ$$

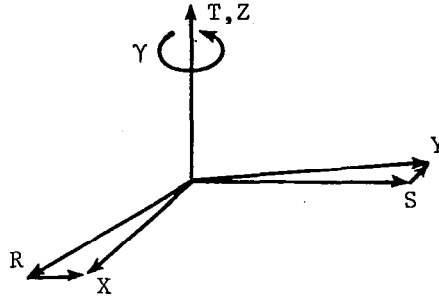
$$\beta = \begin{cases} 23.17^\circ & \text{for orthographic and stereoscopic projections} \\ 0.0^\circ & \text{for perspective projection} \end{cases}$$

$$\alpha = 0.0^\circ$$

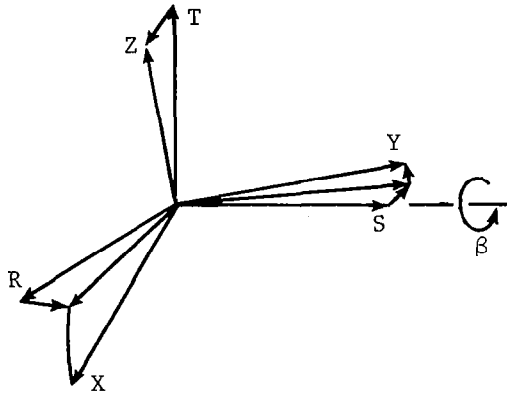
The order in which  $\gamma$ ,  $\beta$ , and  $\alpha$  are specified is critically important, as illustrated in figure 8.

The AXES card can be used to preposition the object in  $90^\circ$  increments in such a manner that only rotations less than  $90^\circ$  are required by the VIEW card to obtain the desired orientation. This is accomplished by entering X, Y, Z, MX, MY, or MZ in the fields corresponding to R, S, or T axes where MX, MY, and MZ represent the negative X-, Y-, and Z-axis directions, respectively. The default values are X, Y, and Z.

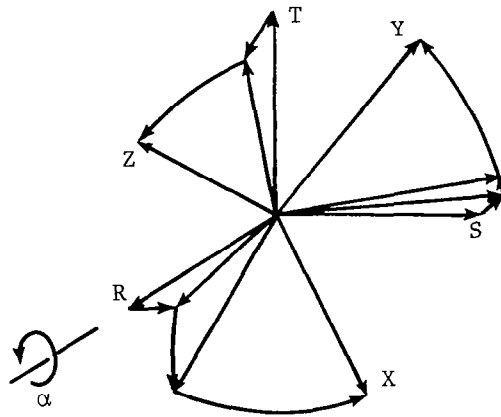
An undeformed or deformed plot of the symmetric portion of an object can be obtained by reversing the sign of the axis that is normal to the plane of symmetry. In the case of multiple planes of symmetry, the signs of all associated planes should be reversed. The ANTISYMMETRIC option should be specified when a symmetric structure is loaded in an unsymmetric manner. This option will cause the deformations to be plotted antisymmetrically with respect to the specified plane or planes. Since the AXES card applies to all parts (SETS) of a single frame, symmetric and antisymmetric combinations cannot be made with this card.



(a)  $\gamma$ -rotation about T-axis.



(b)  $\beta$ -rotation about S-axis.



(c)  $\alpha$ -rotation about R-axis.

Figure 8.- Plotter coordinate system-model orientation.

MAXIMUM DEFORMATION d

This card must always be included if a deformed structure is to be plotted. The value of d represents the length to which the maximum displacement component is scaled in each subcase. The maximum deformation of the structure must be specified in units of the structure (not inches of paper). These data are necessary since the actual deformations are usually too small to be distinguished from the undeformed structure if they are plotted to true scale. If FIND card parameters are to be based on the deformed structure, the FIND card must be preceded by the MAXIMUM DEFORMATION card.

SCALE a[ , b]

- a real number representing scale to which the model is drawn
- b ratio of model size to real object size (stereoscopic projection only)

For orthographic or perspective projections, the scale a is the ratio of the plotted object in inches to the real object in units of the structural model (i.e., one inch of paper equals one unit of structure). For stereoscopic projection, the stereoscopic effect is enhanced by first reducing the real object to a smaller model (scale b) and then applying scale a. The ratio of plotted to real object is then the product  $a \times b$ . A scale must be defined in order to make a plot; however, the SCALE card is not recommended for general use.

ORIGIN i, u, v

- i origin identification number (any positive integer)
- u horizontal displacement of paper origin from RST origin
- v vertical displacement of paper origin from RST origin

In the transformation performed for any of the three projections, the origins of both the object (XYZ system) and of the observer (RST system) are assumed to be coincident.

This card refers to the paper origin. It represents the displacement of the paper origin (lower left-hand corner) from the RST origin. The units are inches and are not subject to the scaling of the plotted object. The ORIGIN card is not recommended for general use.

Ten origins are permitted to be active at one time. However, any one can be redefined at any time. An eleventh origin is also provided if more than ten origins are erroneously defined (i.e., only the last of these surplus

origins will be retained). However, when a new projection or plotter is called for, all previously defined origins are deleted.

VANTAGE POINT  $r_o, s_o, t_o$  [ $s_{or}$ ]

(Perspective and stereoscopic projections only)

- $r_o$  R coordinate of the observer
- $s_o$  S coordinate of the observer in perspective projection or  
S coordinate of the left eye of the observer in the stereoscopic  
projection
- $t_o$  T coordinate of the observer
- $s_{or}$  S coordinate of the right eye of the observer in the stereoscopic  
(not needed in perspective) projection

This card defines the location of the observer with respect to the structural model. A vantage point is required for either perspective or stereoscopic projection. The VANTAGE POINT card is not recommended for general use.

PROJECTION PLANE SEPARATION  $d_o$

(Perspective and stereoscopic projections only)

This card specifies the R-direction separation of the observer and the projection plane.

The PROJECTION PLANE SEPARATION card is not recommended for general use. See the FIND card described at the end of this section. The card may be omitted if VANTAGE POINT is included on the FIND card.

OCULAR SEPARATION  $\left\{ \frac{2.756}{os} \right\}$

(Stereoscopic projection only)

S-coordinate separation of the two vantage points in the stereoscopic projection is defaulted to 2.756 in., which is the separation used in the standard stereoscopic cameras and viewers (70 mm). It is recommended that the default value be used.

CAMERA $\left\{ \begin{array}{c} \text{FILM} \\ \text{PAPER} \\ \text{BOTH} \end{array} \right\} \left[ \text{, BLANK\_FRAMES } \left\{ \begin{array}{c} n \\ \underline{1} \end{array} \right\} \right]$
---

(Microfilm plotters only)

This card offers three options of different cameras or combinations:

FILM - 35-mm or 16-mm film (positive or negative images)

PAPER - positive prints

BOTH - positive prints and 35-mm or 16-mm film

The request for a 35-mm or 16-mm camera and positive or negative images must be communicated to the plotter operator through normal means of communications at the installation. Insertion of blank frames between plots is optional and is applicable only to plots generated on film. The type option must be FILM or BOTH if blank frames are desired. The plotter must be operated in the manual mode in order to have blank frames inserted between positive prints. If blank frames are desired only on film and not on paper, the plotter must be operated in the automatic mode. The default values are Type = PAPER, n = 1. This card is completely optional.

PAPER SIZE $\left\{ \begin{array}{c} a \\ \underline{8.5} \end{array} \right\} \left\{ \begin{array}{c} x \\ \text{BY} \end{array} \right\} \left\{ \begin{array}{c} b \\ \underline{11.0} \end{array} \right\} \left[ \text{, TYPE } \left\{ \begin{array}{c} \text{BCD value} \\ \underline{\text{VELLUM}} \end{array} \right\} \right]$
--

(Table plotters only)

a horizontal size of paper in inches

b vertical size of paper in inches

The default parameters are 8.5 x 11.0, type VELLUM. This card is completely optional.

PEN $\left\{ \begin{array}{c} i \\ \underline{1} \end{array} \right\} \left[ \text{, SIZE } \left\{ \begin{array}{c} j \\ \underline{1} \end{array} \right\} \right] \left[ \text{, COLOR } \left\{ \begin{array}{c} \text{name} \\ \underline{\text{BLACK}} \end{array} \right\} \right]$
--

(Table plotters only)

i pen designation number

j pen size number (0 to 3)

name color desired

This card generates a message on the printed output which may be used to inform the plotter operator as to what size and which color pen point to mount in the various pen holders. The actual number of pens available will depend on the plotter hardware configuration at each installation. This card does not control the pen used in generating the plot. The PEN card is optional and is not appropriate for microfilm plotters.

The pen designations vary on various plotters; therefore, the designation numbers used here are only the pointers to true identification of the pens. The following table summarizes these pen designations and the actual pen numbers on the plotters used:

NASTRAN pen designation	PLOTTER pen number	
	EAI 3500	All others
1	0	1
2	1	2
3	2	3
4	3	4
5	4	1
6	5	2
7	6	3
8	7	4

FIND [SCALE], [ORIGIN i], [VANTAGE POINT], [SET j], [REGION le,be,re,te]

- i origin identification number (any positive integer)
- j set identification number (any positive integer)
- le fractional distance of left edge of plot region from the lower left corner of the image area (Default value = 0)
- be fractional distance of bottom edge of plot region from the lower left corner of the image area (Default value = 0)
- re fractional distance of right edge of plot region from the lower left corner of the image area (Default value = 1)
- te fractional distance of top edge of plot region from the lower left corner of the image area (Default value = 1)

The FIND card requests the structure plotter to compute any of the parameters SCALE, ORIGINi, and/or VANTAGE POINT indicated by the user based on the plotter requested on the PLOTTER card; the projection requested on the PROJECTION card; SETj and REGION le,be,re,te requested on the FIND card; the orientation requested on the VIEW and/or AXES card(s); the deformation scaling requested on the MAXIMUM DEFORMATION card; and the paper size for

table plotters as requested on the PAPER SIZE card. All dependencies on which a FIND card is based must precede the FIND card.

Any one, two, or all three parameters may be computed by the program by using this card, provided that the parameters not requested have already been defined. If no set is specified on this card, the first set defined is used by default. If no options are specified on the FIND card, a SCALE and VANTAGE POINT are selected and ORIGIN 1 is located by using the first defined SET so that the plotted object is located within the image area. The plot region is defined as some fraction of the image area (Image area = 0, 0, 1., 1. and First quadrant = .5, .5, 1., 1.). The image area is located inside the margins on the paper. Each FIND card must be one logical card. The FIND card is recommended for general use.

The following is a diagram of the PLOT execution card:

$$\begin{array}{l}
 \text{PLOT} \left[ \begin{array}{l} \text{STATIC} \\ \text{MODAL} \end{array} \right] \left[ \begin{array}{l} \text{DEFORMATION} \\ \text{VELOCITY} \\ \text{ACCELERATION} \end{array} \right] i_1, i_2 \text{ THRU } i_3, \text{ etc.} \left[ \begin{array}{l} \text{RANGE } f_1, f_2 \\ \text{RANGE } \lambda_1, \lambda_2 \\ \text{TIME } t_1, t_2 \end{array} \right] \left[ \begin{array}{l} \text{PHASE LAG } \phi \\ \text{MAGNITUDE} \end{array} \right], \\
 \\
 \left[ \text{MAXIMUM DEFORMATION } d \right], \\
 \\
 \left[ \text{SET } j_1 \right] \left[ \text{ORIGIN } k_1 \right] \left[ \begin{array}{l} \text{SYMMETRY} \\ \text{ANTISYMMETRY} \end{array} \right] w \left[ \begin{array}{l} \text{PEN} \\ \text{DENSITY} \end{array} \right] p \left[ \text{SYMBOLS } m [, n] \right] \left[ \text{LABEL} \right. \\
 \\
 \left. \begin{array}{l} \text{GRID POINTS} \\ \text{ELEMENTS} \\ \text{BOTH} \end{array} \right] \\
 \\
 \left[ \begin{array}{l} \text{SHAPE} \\ \text{VECTOR } v \\ \text{SHAPE, VECTOR } v \end{array} \right], \\
 \\
 \left[ \text{SET } j_2 \right] \left[ \text{ORIGIN } k_2 \right] \dots \text{ etc.}
 \end{array}$$

This logical card will cause one picture to be generated for each subcase, mode, or time step requested by using the current parameter values. If only the word PLOT appears on the card, a picture of the undeformed structure will be prepared using the first defined set and the first defined origin. The available plot options and their meanings are:

- STATIC            Plot static deformations.
- MODAL            Plot mode shapes.
- DEFORMATION    Nonzero integers following refer to subcases that are to be plotted. Default is all subcases. See SHAPE and VECTOR for use of "0" command.

**VELOCITY** Nonzero integers following refer to subcases that are to be plotted. Default is all subcases.

**ACCELERATION** Nonzero integers following refer to subcases that are to be plotted. Default is all subcases.

**i1,i2,...** Nonzero integers following refer to subcases that are to be plotted. Default is all subcases. See **SHAPE** and **VECTOR** for use of "0" (underlay) command.

**RANGE** Refers to range of eigenvalues or frequencies by using requested subcases for which plots will be prepared.

**TIME** Refers to time interval by using requested subcases and output time steps for which plots will be prepared.

**PHASE LAG** Real number  $\phi$  in degrees (default is 0.0). The plotted value is  $u_R \cos \phi - u_I \sin \phi$  where  $u_R$  and  $u_I$  are the real and imaginary parts, respectively, of the response quantity.

**MAGNITUDE** Plotted value is  $\sqrt{u_R^2 + u_I^2}$ .

**MAXIMUM DEFORMATION** Real number following is used as the maximum displacement component in scaling the displacements for all subcases. Each subcase is separately scaled according to its own maximum if this item is absent.

**SET** Integer following identifies a set which defines the portion of the structure to be plotted. Default is first set defined.

**ORIGIN** Integer following identifies the origin to be used for the plot. Default is first origin defined.

**SYMMETRY w** Prepare an undeformed or deformed plot of the symmetric portion of the object which is defined by **SET j**. This symmetric portion will be located in the space adjacent to the region originally defined in **ORIGIN k** and will appear as a reflection about the plane whose normal is oriented parallel to the coordinate direction **w**.

**ANTISYMMETRY w** Prepare a deformed plot of the symmetric portion of the antisymmetrically loaded object which is defined by **SET j**. This symmetric portion will be located in the space adjacent to the region originally defined by **ORIGIN k** and will appear as a reflection of the antisymmetrically deformed structure about the plane whose normal is oriented parallel to the coordinate direction **w**.

The symbol **w** may specify the basic coordinates **X**, **Y**, or **Z** or any combination of the three. This option allows the plotting of symmetric and/or antisymmetric combinations provided that an origin is selected for



the portion of the structure defined by the bulk data that allows sufficient room for the complete plot. This does not permit the combination of symmetric and antisymmetric subcases, as each plot must represent a single subcase. In the case of a double reflection, the figure will appear as one reflected about the plane whose normal is parallel to the first of the coordinates  $w$ , followed by a reflection about the plane whose normal is oriented parallel to the second of the coordinates  $w$ . This capability is primarily used in the plotting of structures that are loaded in a symmetric or an antisymmetric manner. The plane of symmetry must be one of the basic coordinate planes.

**PEN** Integer following controls the internal NASTRAN pen number that is used to generate the plot on table plotters.

**DENSITY** Integer following specifies line density for film plotters. A line density of  $d$  is  $d$  times heavier than a line density of 1.

**SYMBOLS  $m[,n]$**  All of the grid points associated with the specified set will have symbol  $m$  overprinted with symbol  $n$  printed at its location. If  $n$  is not specified, only symbol  $m$  will be printed. Grid points excluded from the set will not have a symbol. Grid points in an undeformed underlay will be identified with symbol 2.

The following is a table of symbols available on each plotter. Symbols that are not available on a given plotter are defaulted to a similar symbol indicated in parentheses.

Symbol		Availability		
No. $m$ or $n$	Symbol	EAI 3500	SC 4020	All others
0	No symbol	X	X	X
1	X	X	X	X
2	*	X	X	X
3	+	X	X	X
4	-	X	X	X
5	.	X	X	X
6	○	X	X	X
7	□	X	X	X
8	◇	X	(7)	X
9	△	(7)	(7)	X

**LABEL GRID POINTS** All the grid points associated with the specified set have their identification number printed to the right of the undeflected or deflected location (undeflected location in the case of superimposed plots).

**LABEL ELEMENTS** All the elements included in the specified set are identified by the element identification number and type at the center of each element (undeflected location in the case of superimposed plots).

**LABEL BOTH** Label both the grid points and elements.

Labels for element types are given in the following table:

BAR - BR  
SHEAR - SH  
TRIA2 - T2  
TRMEM - TM

**SHAPE** All the elements included in the specified set are shown by connecting the associated grid points in a predetermined manner.

Both deformed and undeformed shapes may be specified. All of the deformed shapes relating to the subcases listed may be underlaid on each of their plots by including "0" with the subcase string on the PLOT card. The undeformed plot will be drawn using PEN 1 or DENSITY 1 and SYMBOL 2 (if SYMBOLS is specified).

**VECTOR v** A line will be plotted at the grid points of the set to represent the deformation of the point in length and direction.

Vectors representing the total deformation or its principal components may be plotted by insertion of the proper letter(s) for variable v. Possible vector combinations are:

X or Y or Z	Requesting individual components
XY or XZ or YZ	Requesting 2 specified components
XYZ	Requesting all 3 components
RXY or RXZ or RYZ	Requesting vector sum of 2 components
R	Requesting total vector deformation
N	Used with any of the above combinations to request no underlay shape be drawn

All plots requesting the VECTOR option shall have an underlay generated of the undeformed shape using the same sets, "PEN 1" or "DENSITY 1," and symbol 2 (if SYMBOLS is specified). If "SHAPE" and "VECTOR" are specified, the underlay will depend on whether "0" is used with DEFORMATION. It will be the deformed shape when not used and will be both deformed and undeformed shapes when it is used. The part of the vector at the grid point will be the tail when the underlay is undeformed and the head when it is deformed. If the "N" parameter is used, no shape will be drawn but other options such as SYMBOLS will still be valid.

### 3.2.7 Summary of Structure Plot Request Packet Cards

#### SET Definition - required

SET i [INCLUDE] [ELEMENTS] j <sub>1</sub> , j <sub>2</sub> , j <sub>3</sub> THRU j <sub>4</sub> , j <sub>5</sub> , etc.					
<table style="border: none;"> <tr> <td style="border: 1px solid black; padding: 2px;">[INCLUDE]</td> <td rowspan="3" style="border: none; padding: 0 10px;">[ELEMENTS GRID POINTS]</td> <td rowspan="3" style="border: none;">k<sub>1</sub>, k<sub>2</sub>, k<sub>3</sub> THRU k<sub>4</sub>, k<sub>5</sub>, etc.</td> </tr> <tr> <td style="border: none; padding: 2px;">[EXCLUDE]</td> </tr> <tr> <td style="border: none; padding: 2px;">[EXCEPT]</td> </tr> </table>	[INCLUDE]	[ELEMENTS GRID POINTS]	k <sub>1</sub> , k <sub>2</sub> , k <sub>3</sub> THRU k <sub>4</sub> , k <sub>5</sub> , etc.	[EXCLUDE]	[EXCEPT]
[INCLUDE]	[ELEMENTS GRID POINTS]			k <sub>1</sub> , k <sub>2</sub> , k <sub>3</sub> THRU k <sub>4</sub> , k <sub>5</sub> , etc.	
[EXCLUDE]					
[EXCEPT]					

#### Parameter Definition - optional, except as noted

PLOTTER plotter name, MODEL name [ DENSITY { 800 556 200 } BPI ]	(Required if not SC-4020)
--	---------------------------

<table style="border: none;"> <tr> <td style="border: none; padding: 2px;">{ ORTHOGRAPHIC PERSPECTIVE STEREOSCOPIC }</td> <td style="border: none; padding: 0 10px;">PRJECTION</td> </tr> </table>	{ ORTHOGRAPHIC PERSPECTIVE STEREOSCOPIC }	PRJECTION
{ ORTHOGRAPHIC PERSPECTIVE STEREOSCOPIC }	PRJECTION	

AXES r, s, t [ { SYMMETRIC ANTISYMMETRIC } ]
---

VIEW γ, β, α
--------------

SCALE a [ , b ]	(Required if not on FIND card)
-----------------	--------------------------------

ORIGIN i, u, v	(Required if not on FIND card)
----------------	--------------------------------

VANTAGE POINT r <sub>0</sub> , s <sub>0</sub> , t <sub>0</sub> [ , s <sub>0r</sub> ]	(Required for perspective and stereoscopic projections if not on FIND card)
--	---

PROJECTION PLANE SEPARATION d <sub>0</sub>	(Required for perspective and stereoscopic projections if VANTAGE POINT not on FIND card)
--	---

OCULAR SEPARATION { $\frac{2.756}{os}$ }
--

MAXIMUM DEFØRMATION d

(Required if deformed slopes are to be drawn)

PEN  $\left\{ \begin{array}{c} i \\ \underline{1} \end{array} \right\}$  [ , SIZE  $\left\{ \begin{array}{c} j \\ \underline{1} \end{array} \right\}$  ] [ , CØLØR  $\left\{ \begin{array}{c} \text{name} \\ \text{BLACK} \end{array} \right\}$  ]

CAMERA  $\left\{ \begin{array}{c} \text{FILM} \\ \text{PAPER} \\ \text{BØTH} \end{array} \right\}$  [ , BLANK FRAMES  $\left\{ \begin{array}{c} n \\ \underline{1} \end{array} \right\}$  ]

PAPER SIZE  $\left\{ \begin{array}{c} a \\ \underline{8.5} \end{array} \right\}$   $\left\{ \begin{array}{c} x \\ \text{BY} \end{array} \right\}$   $\left\{ \begin{array}{c} b \\ \underline{11.0} \end{array} \right\}$  [ , TYPE  $\left\{ \begin{array}{c} \text{BCD Value} \\ \text{VELLUM} \end{array} \right\}$  ]

FIND Card - optional

FIND [SCALE], [ØRIGIN i], [VANTAGE PØINT], [SET j], [REGIØN le, be, re, te]

PLØT Execution Card - required

PLØT  $\left[ \left\{ \begin{array}{c} \text{STATIC} \\ \text{MØDAL} \end{array} \right\} \right] \left[ \left\{ \begin{array}{c} \text{DEFØRMATION} \\ \text{VELOCITY} \\ \text{ACCELERATION} \end{array} \right\} \right] i_1, i_2 \text{ THRU } i_3, \text{ etc. } \left[ \left\{ \begin{array}{c} \text{RANGE } f_1, f_2 \\ \text{RANGE } \lambda_1, \lambda_2 \\ \text{TIME } t_1, t_2 \end{array} \right\} \right] \left[ \left\{ \begin{array}{c} \text{PHASE LAG } \phi \\ \text{MAGNITUDE} \end{array} \right\} \right],$

[MAXIMUM DEFØRMATION d],

[SET j1] [ØRIGIN k1]  $\left[ \left\{ \begin{array}{c} \text{SYMMETRY} \\ \text{ANTISYMMETRY} \end{array} \right\} w \right] \left[ \left\{ \begin{array}{c} \text{PEN} \\ \text{DENSITY} \end{array} \right\} p \right] \left[ \text{SYMBØLS } m [ , n ] \right] \left[ \text{LABEL} \right]$

$\left[ \begin{array}{c} \text{GRID PØINTS} \\ \text{ELEMENTS} \\ \text{BØTH} \end{array} \right]$

$\left[ \begin{array}{c} \text{SHAPE} \\ \text{VECTØR } v \\ \text{SHAPE, VECTØR } v \end{array} \right],$

[SET j2] [ØRIGIN k2] .... etc.

### 3.3 NASTRAN General Purpose Plotter

One feature which the NASTRAN plotting software lacks is the ability to directly use the plotting equipment attached on-line to a computer. This is not due to special purpose programming but rather to one of the basic characteristics of NASTRAN, computer independence. To access on-line plotters would not only make NASTRAN computer dependent, but probably installation dependent also. This installation dependency would result from the necessity of using special subroutines provided by the computer installation to access the on-line plotter, with no guarantee that subroutines having the same name and calling sequences would be available at any other computer installation. Even so, there would almost certainly occur a subroutine-naming conflict due to the great number of subroutines in NASTRAN.

An effort is made in NASTRAN to partially overcome this deficiency. In general, NASTRAN will produce a plot tape which can be used directly by any one of several off-line plotters. In addition, NASTRAN can be directed by the user to produce a so-called "general purpose plotter" tape. Another program completely external to NASTRAN would then have to exist, its function being to translate this plot tape for the on-line plotter so that it will produce the plots intended by NASTRAN. This implies that in order to produce a NASTRAN plot, two programs must be run: first, NASTRAN itself, and then the external translator program.

The purpose of this section is to explain the characteristics and construction of the NASTRAN general purpose plotter tape so that a programmer will be able to write a program to translate this plot tape for the on-line plotter. Understanding the overall logic used by the NASTRAN plotter software package in producing a plot tape will simplify the task of writing this translator program. It is therefore recommended that the programmer become familiar with this section.

The NASTRAN general purpose plotter tape is composed of a simple set of elementary plot operations which can easily be deciphered by a FORTRAN program on any digital computer. As each operation is deciphered, the translator program should direct the on-line plotter to appropriate action. This would normally be done by using the installation software to interface between the translate program and the on-line plotter. With the existence of this external translator program, NASTRAN would then have the capability of indirectly referencing the corresponding on-line plotter. A by-product of this environment is the implied capability of indirectly accessing any plotter, whether on-line or off-line, after the appropriate external translator programs are written.

The NASTRAN general purpose plotter tape is a seven-track, odd parity, fixed-length record tape. An end-of-file mark follows the last plot only. Each record is composed of 3000 six-bit-unsigned integers (750 words on an IBM 360, 500 words on a UNIVAC 1108, and 300 words on a Control Data 6600 computer system) and is composed of 100 plot commands, each being composed of

30 six-bit unsigned integers (15 half-words on an IBM 360, 5 words on a UNIVAC 1108, and 3 words on a Control Data 6600 computer system). Not all plot commands will have useful information in all 30 six-bit integers. Some commands use only 2 of the 30 six-bit integers while others use 22. The general format of each command is as follows:

$$PCR_4R_3R_2R_1R_0S_4S_3S_2S_1S_0T_4T_3T_2T_1T_0U_4U_3U_2U_1U_000000000$$

where

P            plot command  
 C            control index  
 R<sub>i</sub>          decimal digit of an integer called R  
 S<sub>i</sub>          decimal digit of an integer called S  
 T<sub>i</sub>          decimal digit of an integer called T  
 U<sub>i</sub>          decimal digit of an integer called U  
 0            zero

The plot command is a six-bit integer, any one of seven possible plot commands, as follows:

0            no operation  
 1            start new plot  
 2            select camera  
 3            skip to a new frame  
 4            type a character (may also equal 14)  
 5            draw a line (may also equal 15)  
 6            draw an axis (may also equal 16)

The control index is also a six-bit integer. It may be a pen number, a line density, a camera number, or a pointer into a list of characters and symbols. The four integer values (R,S,T,U) specified in a command must be reconstructed by the external translator program. Each integer value is represented in the command as follows:

$$d_4d_3d_2d_1d_0$$

where the original integer value is given by:

$$d_4 10^4 + d_3 10^3 + d_2 10^2 + d_1 10^1 + d_0 10^0$$

The significance of each of the four integer values (R,S,T,U) may vary from one plot command to another.

Command 0, no operation, is simply a padding for plot records that may otherwise have been less than 300 characters long. All 30 characters of this command will be zero.

Command 1, start new plot, will always be the first command introducing each new plot. The first integer, R, will be the plot number. The second and third integers, S and T, are the maximum x and y values specified in any other command for this plot. The minimum x and y values are always zero and are therefore not specified in this command. If necessary, the translator program can use these maximum x and y values to scale subsequent integer values so that the plot will not exceed the limits of the plotting surface. The plot number is included because some plotters require the plot number as part of the first command for each new plot. In addition, if the receiving plotter is a table plotter, the translator program should issue a command to the plotter which will stop it so that the plotter operator can change the paper. If the plotter is a drum plotter, the translator program must skip a sufficient amount of paper to insure that the previous plot will not be over-plotted. And if the receiving plotter is a microfilm plotter, nothing else need be done.

Command 2, select camera, uses only the control index C. The remaining 28 characters are always zeros. This command is meaningful only on a microfilm plotter having both film and hardcopy output. The control index is the camera or medium request number: 1 is film only, 2 is hardcopy (paper) only, and 3 is both. Upon receiving this command, the translator program should issue a command to the receiving plotter to select the requested camera or output medium; then this command should be ignored.

Command 3, skip to a new frame, also uses only the control index. The remaining 28 characters are always zeros. This command is meaningful only on a microfilm plotter. The control index is the camera or output medium request number: 1 is film only, 2 is hardcopy (paper) only, and 3 is both. The appropriate camera will have already been selected in a previous command. The only reason the camera number is included in this command is because some microfilm plotters require the camera or output medium to be specified in both a select-camera and skip-frame command. Upon receiving this command, the translator program should issue a command to the receiving plotter to skip to a new frame. If the receiving plotter is not a microfilm plotter, then this command should be ignored. Note: At least one command 3 will appear after each command 1 and before the next command 1.

Command 4, type character; command 5, draw line; and command 6, draw axis, will always occur in sets (i.e., a set of command 4, a set of command 5, a set of command 6). There may be more than one set of each type of command, but within a set, the commands will all be of an identical type. This is done because on some plotters it is very inefficient to frequently change modes of operation (e.g., typing mode, line drawing mode). The plot command of the first command in a set will always be equal to ten plus the basic plot command value (i.e., command 4 equals 14; command 5 equals 15, and command 6 equals 16). In all subsequent plot commands in the set, the plot command value will always equal the basic plot command value.

For command 4, the control index is a pointer into a specific list of characters and special symbols. The list of characters to which the pointer applies is shown at the end of this section. The first two integer values (R and S) in the plot command represent the x and y coordinates of the point on the plotting surface at which the center of the character or symbol should be typed. The remaining 18 characters of the command are always zeros. Upon receipt of command 4, the translator program should issue a command to the receiving plotter to type the requested character or special symbol at the specified point. Of course, there is no guarantee that all the possible characters and special symbols can be typed by the receiving plotter. If any character or special symbol cannot be typed by the receiving plotter, the translator program will then have to make a substitution or not type the character at all.

For command 5, the control index is either a pen number (for table and drum plotters) or a line density (for microfilm plotters). If the receiving plotter is a microfilm plotter, it is recommended that the translator program simply draw the line as many times as is indicated by the line density value rather than use any special density settings available on the plotter hardware. The first two integer values (R and S) represent the x and y coordinates of the starting point of the line. The next two integer values (T and U) represent the x and y coordinates of the ending point of the line. The last eight characters of the command are always zeros. Upon receiving this command, the translator program should issue a command to the receiving plotter to draw the line. However, some plotters require that a line be broken into a series of short lines. If this is the case on the receiving plotter, the translator program will have to accomplish this task unless the installation software makes provision for this automatically.

Command 6 is identical to command 5. The only difference is in the orientation of the drawn line. The line drawn by command 6 will always be either horizontal or vertical. For most plotters, the translator program will handle this command just like command 5. However, some plotters which would ordinarily require that lines be broken into a series of short lines may have a special command available to draw a horizontal or vertical line of any length. Only for these few plotters will this command have any special significance in the translator program. If such is the situation, the translator program, upon receiving this command, should issue a command to the receiving plotter to



draw the axis. Otherwise, the translator program should simply issue a command to the receiving plotter to draw a line representing the axis.

The sequence of characters is

0	1	2	3	4	5	6	7	8	9
A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z	(	)	+	-
*	/	=	.	,	\$	'		dot	circle
									blank

Square diamond triangle (point up)

#### 4. DIRECT MATRIX ABSTRACTION PROGRAMMING

##### 4.1 General DMAP Rules

DMAP is the user-oriented language used by NASTRAN to solve problems. DMAP, like English or FORTRAN, has many grammatical rules which must be followed in order to be interpreted by the NASTRAN DMAP compiler. Section 4.2 provides the user with the rules of DMAP that allow him to construct a DMAP sequence by using the many modules contained in the NASTRAN DMAP repertoire.

Section 4.1 is an index of the matrix, structural, utility, and executive DMAP modules that are contained in sections 5.2 to 5.5, respectively. The appendix provides several examples of DMAP usage.

##### 4.2 DMAP Rules

Grammatically, DMAP instructions consist of two types, executive operation instructions and functional module instructions. Grammatical rules for these two types of instructions are discussed separately.

Functional modules are arbitrarily classified as structural modules, matrix operation modules, utility modules, or user-generated modules.

The DMAP sequence itself consists of a series of DMAP instructions or statements, the first of which is BEGIN or XDMAP and the last of which is END. The remaining statements consist of executive operation instructions and functional module calls.

### 4.3 DMAP Rules for Functional Module Instructions

The primary characteristic of the functional module DMAP instruction is its prescribed format. The general form of the functional module DMAP statement is:

```
MØD I1,I2,---,Im/01,02,---,On/a1,b1,p1/a2,b2,p2----/az,bz,pz $
```

where

MØD                    DMAP functional module name

Ii; i = 1,m            input data block names

Oi; i = 1,n            output data block names

ai,bi,pi; i = 1,z      parameter sections

In the general form shown, commas separate several like items and slashes separate sections from one another. The module name is separated from the rest of the instruction by a blank or a comma. The dollar sign is used to end the instruction and is not required unless the instruction ends in the delimiter / . Blanks may be used in conjunction with any of the previous delimiters for ease in reading.

A functional module communicates with other modules and with the executive system entirely through its inputs, outputs, and parameters. The module name is a BCD value that consists of an alphabetic character followed by up to seven additional alphanumeric characters. A data block name may be either a BCD value or null. The absence of a BCD value indicates that the data block is not needed for a particular application.

Each functional module DMAP statement must conform to the module properties list (MPL) in the following ways:

Name spelling

Number of input data blocks

Number of output data blocks

Number of parameters

Type of each parameter

See sections 4.4 and 4.5 for allowable exception to these rules.

#### 4.4 Functional Module Input Data Blocks

In most cases, an input data block should have been defined in a DMAP program before it is used. However, there may be instances in which a module can handle, or may even expect, a data block to be undefined at the time the module is initially called. An input data block is previously defined if it appears as an output data block in a previous DMAP instruction, as output from the input file processor, or as any user-input (by means of bulk data cards) DMI data block name. Although the number of data blocks is prescribed, if any number of final data blocks are null, they may be omitted from the section. For example, the module MATPRN which uses five input data blocks may be defined by

```
MATPRN GEØM1,,,, // $
```

```
MATPRN GEØM1 // $
```

A potentially fatal error message (see section 4.8) is issued at compilation time to warn the user that a discrepancy in the data block name list has been detected or that a previously undefined data block is used as input. There is also an error-level option on the XDMAP compiler option card which may be invoked by the user to terminate execution in the event of such errors.

#### 4.5 Functional Module Output Data Blocks

In general, a data block name appears as output only once. However, there are cases in which an output data block may be of no subsequent use in a DMAP program. In such a case, the name may be used again but caution should be used when employing such techniques. Although the number of output data blocks is prescribed, the data block name list may be abbreviated in the manner of section 4.4. Potentially fatal error messages warn the user if possible ambiguities can occur from these usages.

#### 4.6 Functional Module Parameters

Parameters may serve many purposes in a DMAP program. They may pass data values in and/or out of a module or they may be used as flags to control the computational flow within the module or the DMAP program. There are two allowable forms of the parameter section of the DMAP instruction. The first explicitly states the attributes of the parameters, and the second is a briefer simplified specification. The general form of the formal parameter section is

```
/ ai,bi,pi /
```

where the allowable parameter specifications are:

$a_i =$	{	<p>V     Parameter value is variable and may be changed by module during execution.</p> <p>C     Parameter value is prescribed initially by user and is an unalterable constant.</p> <p>S     Parameter is of type V and will be saved automatically at completion of module.</p>
$b_i =$	{	<p>Y     Initial parameter value may be specified on PARAM bulk data card.</p> <p>N     Initial parameter value may not be specified on PARAM bulk data card.</p>
$p_i =$	{	<p>PNAME = v     PNAME is BCD name selected by the user to represent given parameter.</p> <p>PNAME</p> <p>v</p>

The default values for  $a_i$  and  $b_i$  depend on the value given for  $p_i$ . The three forms available for  $p_i$  require additional clarification. The symbol  $v$  represents an actual numeric value for the parameter and may be used only when  $a_i = C$  and  $b_i = N$ . The other forms are clarified by the examples found at the end of this section. Each parameter has an initial value that is established when the DMAP sequence is compiled during execution of the NASTRAN preface. The means by which initial values are established for all DMAP parameters is explained by the symbolic examples that follow. The value used at execution time may differ from the initial value if and only if the module changes the value.

The formal parameter specifications defined above can, in frequently encountered instances, be greatly simplified. Situations where these simplifications may be used are:

/ C,N,v / can be written as / v /

The value  $v$  is written exactly as it would be in the formal specification with the exception of BCD constant parameters, in which case the BCD string is enclosed by asterisks (i.e., / \*STRING\* /).

/ V,N,PNAME / can be written as / PNAME /

/ V,N,PNAME = v / can be written as / PNAME = v /

Again, in the case where the value v appears, it is written exactly as in the case of the formal specification. In this case, BCD strings are not delimited by asterisks.

/ (default value) / can be written as //

If a particular parameter has a predefined default value and the user wishes to choose this value, then it is necessary only to code successive slashes. If a parameter does not have a default value, an error message is issued.

Six parameter types are available and the type of each parameter may not be changed. The types and examples of values as they would be written in DMAP are given as follows:

Parameter type	Value examples		
Integer	7	-2	0
Real	-3.6	2.4+5	0.01-3
BCD	VARØ1	STRING3	B3R56
Double precision	2.5D-3	1.354D7	
Complex single precision		(1.0,-3.24)	
Complex double precision		(1.23D-2,-3.67D2)	

Many possible forms of the parameter section may be used. The following examples help clarify the possibilities:

// Equivalent to / C,N,v / where v is the default value which must exist.

/ C,Y,v Constant input parameter, for example

/ C,N,O / C,N,BKLØ / C,N,(1.0,-1.0)

/ O / \*BKLØ\* / (1.0,-1.0) '

In the examples shown, both in formal and simplified form, the values 0 (integer), BKLØ (BCD), and 1.0-il.0 (complex single precision) are defined.

/ C,Y,NAME                   Constant input parameter; the default value is used unless a PARAM bulk data card referencing PNAME is present. Error condition is detected if either no PARAM card is present or if no default value exists.

/ C,Y,PNAME = v               Constant input parameter; the value v is used unless a PARAM bulk data card referencing PNAME is present

/ V,Y,PNAME  
   ØR                           Variable parameter; may be input, output, or both; initial value is the first of

/ V,Y,PNAME = v               Value from the most recently executed S parameter, if any

                              Value from PARAM bulk data card referencing PNAME will be used if present in bulk data deck

                              v, if present in DMAP instruction

                              MPL default value, if any

                              0

/ V,N,PNAME                   Variable parameter; may be input, output, or both; initial value is the first of

  ØR

/ PNAME                        Value from the most recently executed SAVE instruction, if any

  ØR

/ V,N,PNAME = v               v, if present in DMAP instruction

  ØR

/ PNAME = v                    MPL default value, if any

                              0

If any of the previous V parameters are to be carried forward on S, it must replace the V for that parameter (i.e., /S,N,PNAME).

#### 4.7 DMAP Compiler Options - XDMAP Instruction

The user can elect several options when compiling and executing a DMAP program by including an XDMAP compiler option instruction in the program. The available options are:

GØ (default) or NØGØ

The GØ option compiles and executes the program, and NØGØ terminates the job at the conclusion of compilation.

LIST or NOLIST (default)

This option produces a DMAP program source listing.

DECK or NODECK (default)

This option will produce a punched card deck of the program.

ØSCAR or NØØSCR (default)

If the ØSCAR option is selected, a complete listing of the operation sequence control array will be given.

REF or NØREF (default)

This option will produce a complete cross reference of variable parameters, data block names, and module calls for the DMAP program.

ERR=0 or 1 (default) or 2

This option specifies the error level - '0' for WARNING, '1' for POTENTIALLY FATAL, and '2' for FATAL ERROR MESSAGE - at which termination of the job occurs. See section 4.8 for further explanation.

The complete description of the XDMAP card may be found in the DMAP module description section. An XDMAP card need not appear when all default values are elected but may be replaced with a BEGIN instruction.

#### 4.8 Extended Error Handling Facility

There are three levels of error messages generated during the compilation of a DMAP sequence. These levels are WARNING MESSAGE, POTENTIALLY FATAL ERROR MESSAGE, and FATAL ERROR MESSAGE. The user has, through available compiler options, the ability to specify the error level at which the job will be terminated. (See section 4.7 for the manner of specification.) The class of POTENTIALLY FATAL ERROR MESSAGES is generated by certain compiler conveniences which, if not fully understood by the user, could cause an erroneous or incorrect execution of the DMAP sequence. The default value for the error level is that of the POTENTIALLY FATAL ERROR. An experienced DMAP user often uses compiler conveniences when they are fully understood and executes with the ERR=2 (FATAL ERROR) option.

#### 4.9 DMAP Rules for Executive Operation Instructions

Each executive operation statement has its own format, which is generally open-ended; this means that the number of inputs, outputs, and parameters is

not prescribed. Executive operation instructions or statements are divided into general categories as follows:

Declarative instructions FILE, LABEL, and XDMAP which aid the DMAP compiler and the file allocator, as well as provide user convenience

Instructions EQUIV and PURGE which aid the NASTRAN executive system in allocating files and interfacing between functional modules

Control instructions REPT, JUMP, COND, EXIT, and END which control the order in which DMAP instructions are executed

The rules associated with the executive operation instructions are distinct for each instruction and are discussed individually in section 4.2.

#### 4.10 Techniques and Examples of Executive Module Usage

Before reading this section, the new DMAP user should read sections 5.2 to 5.5 to become familiar with the terminology.

Even though the DMAP program may be interpretable by the DMAP compiler, there is no guarantee that the program will yield the desired results. Therefore, this section is provided to acquaint the DMAP programmer with techniques and examples used in writing DMAP programs. In particular, the instructions REPT, FILE, EQUIV, and PURGE are discussed in some detail.

The data blocks and functional modules referenced in the following examples are fictitious and have no relationship to any real data blocks or functional modules. A data block is described as having a status of "not generated," "generated," or "purged." A status of not generated means that the data block is available for generation by appearing as output in a functional module. A status of generated means that the data block contains data which are available for input to a subsequent module. A status of purged means that the data block cannot be generated and any functional module attempting to use this data block as input or output is informed that the purged data block is not available for use.

#### 4.11 REPT and FILE Instructions

DMAP instructions bounded by both the REPT instruction and the label referenced by the REPT instruction are referred to as a loop. The location referenced by the REPT is called the top of the loop. In many respects, a DMAP loop is like a giant functional module since it requires inputs and generates output data blocks without any special action by the DMAP programmer. The one exception is a data block that is not referenced outside the loop (i.e., an internal data block with respect to the loop). The file allocator considers internal



data blocks as scratch data blocks to be used for the present pass through the loop but not saved for input at the top of the loop. Should the DMAP programmer desire to save an internal data block, he may do so by declaring the data block SAVE in the FILE instruction.

When the REPT instruction transfers control back to the top of the loop, the status of all internal data blocks is changed to not generated unless the internal data block is declared SAVE or APPEND in a FILE instruction. Equivalences established between internal data blocks (not declared saved) and data blocks referenced outside the loop are not carried over the next time through the loop. The equivalence must be reestablished each time through the loop. Data blocks generated by the input file processor are considered referenced outside all DMAP loops. The following example shows the use of REPT and FILE instructions:

DMAP {	XDMAP	ERR=2
	FILE	X=SAVE / Y=APPEND / Z=APPEND \$
	LABEL	L1 \$
	MØD1	B/W,Y \$
	CØND	L3,PX \$
	MØD2	A/X/S,N,PX=0 \$
	LABEL	L3 \$
	MØD3	W,X,Y/Z \$
	REPT	L1,1 \$
	MØD4	Z// \$
	END	\$

Assume that MØD2 sets PX = 0 when it is executed. Note that Z is declared APPEND whereas Y will be saved since it is an internal data block that is to be appended. X is an internal data block that is to be saved since it is only generated the first time through the loop but is needed as input each time the loop is repeated. W is an internal data block that is generated each time through the loop; therefore, it is not saved. This example is controlled with a constant REPT instruction. To code loops where the loop variable is unknown until execution time, the variable REPT option may be used.

The following table shows what happens when the preceding DMAP program is executed. Only modules being executed are shown in the table. Data blocks A and B are assumed to be generated by the input file processor and hence are considered referenced outside all DMAP loops.

Module being executed	Input status and comments	Output status and comments
MØD1	B - assumed generated by the input file processor	W,Y - generated
CØND	PX is 0	No transfer occurs since $PX \geq 0$
MØD2	A - assumed generated by the input file processor	X - generated PX is set $< 0$ and saved for subsequent use
MØD3	W,X,Y are all generated at this point	Z - generated
REPT	Loop count is initially set at 1	Transfer to L1 - set loop count to $1 - 1 = 0$ ; status of data blocks at top of loop will be A,B,Z - generated (referenced outside loop); X,Y - generated (internal data blocks declared saved) W - not generated (internal data block)
MØD1	B - generated	W - generated Y - generated (appended)
CØND	PX is not $< 0$ due to SAVE	Transfer to L3 occurs
MØD3	W,X,Y - generated	Z - generated (appended)
REPT	Loop count is now 0	No transfer occurs
MØD4	Z - generated	Output to printer (assumed)
END		Normal termination of problem

#### 4.12 EQUIV Instruction

There are no restrictions on the status of data blocks referenced in an EQUIV instruction. Consider the instruction EQUIV A,B<sub>1</sub>,---,B<sub>N</sub>/P \$ when  $P < 0$ . Data blocks B<sub>1</sub>,---,B<sub>N</sub> take on all the characteristics of data block A including the status of A; thus, the status of some B<sub>j</sub> can change from purged to generated or not generated.

The EQUIV instruction will unequivalence data blocks when  $P \geq 0$ . In an unequivalence operation, the status of all secondary data blocks reverts to not generated. Suppose A, B, and C are all equivalenced and  $P \geq 0$ . EQUIV A,B/P \$ breaks the equivalence between A and B but not between A and C.

Now consider the following situation: Data block B is to be generated by repeatedly executing functional module MØD2. The input to MØD2 is the previous output from MØD2. That is to say, each successive generation of B depends on the previous B generated. The following example shows how the EQUIV

instruction is used to solve this problem. (Assume parameter  $BREAK \geq 0$  and parameter  $LINK < 0$ .)

```

      XDMAP      ERR=2 $
      MØD1      A/B $
      LABEL    L1 $
      EQUIV    B, BB/BREAK $
      MØD2      B/BB $
      EQUIV    BB, B/LINK $
      REPT     L1, 1 $
      MØD3      BB// $
      END      $
  
```

The following table shows what happens when the above DMAP program is executed. Only modules being executed are shown in the table.

Module being executed	Input status and comments	Output status and comments
MØD1	A - assumed generated by input processor	B - generated
EQUIV	B will not be equivalenced to BB since $BREAK \geq 0$	No action taken
MØD2	B - generated	BB - generated
EQUIV	BB and B are not equivalenced; B - generated; BB - generated $LINK < 0$	B is equivalenced to BB; that is, B assumes all of the characteristics of BB and B and BB then both have the status of generated
REPT	Loop count is initially 1	Transfer to L1; set loop count to $1 - 1 = 0$
EQUIV	B and BB are generated and equivalenced; $BREAK \geq 0$	The equivalence is broken; B - generated; BB - not generated
MØD2	B - generated	BB - generated
EQUIV	BB and B are generated and not equivalenced; $LINK < 0$	B equivalenced to BB; B, BB - generated
REPT	Loop count is 0	No transfer occurs
MØD3	BB - generated	Output to printer (assumed)
END		Normal termination of problem

Since equivalences are automatically broken between internal files not declared saved and files referenced outside the loop, the above DMAP program could be written as follows and the same results achieved:

```

          XDMAP      ERR=2 $
          MOD1      A/B $
          LABEL     L1 $
DMAP loop { MOD2      B/BB $
          EQUIV     BB,B/LINK $
          REPT      L1,1 $
          MOD3      B// $
          END       $

```

Data block BB is not internal; therefore, the instruction EQUIV B,BB/BREAK \$ is not needed.

#### 4.13 PURGE Instruction

The status of a data block is changed to purged by explicitly or implicitly purging it. A data block is explicitly purged through the PURGE instruction whereas it is implicitly purged if it is not created by the functional module in which it appears as an output.

The primary purpose of the PURGE instruction is to prepurge data blocks. Prepurging is the explicit purging of a data block prior to its appearance as output from a functional module. Prepurging data blocks allows the NASTRAN executive system to allocate available files more efficiently and decreases problem execution time. The DMAP programmer should look for data blocks that can be prepurged and purge them as soon as it is recognized that they are not generated.

Sometimes during the execution of a problem it is necessary to generate a data block whose status is purged. This situation can occur in DMAP looping. To generate a data block that is purged, it is first necessary to unpurge it (i.e., change its status from purged to not generated). Unpurging is achieved by executing a PURGE instruction which references the purged data block and whose purge parameter is positive.

The PURGE instruction thus has two functions, to unpurge and purge data blocks, depending on the value of the purge parameter and the status of the referenced data block. The following table shows that action is taken by the PURGE instruction for all combinations of input:

PURGE A/P \$		
Status of data block prior to PURGE	Value of P	Status of data block A after PURGE
Not generated	$P \geq 0$	Not generated (i.e., no action taken)
Not generated	$P < 0$	Purged
Generated	$P \geq 0$	Generated (i.e., no action taken)
Generated	$P < 0$	Purged
Purged	$P \geq 0$	Not generated (i.e., unpurged)
Purged	$P < 0$	Purged (i.e., no action taken)

The user may wonder why he should not prepurge all data blocks and then un purge them when necessary in order to really assist the file allocator. This is not recommended, however, since there is a limited amount of space in the table where the status of data blocks is kept. It is possible to overflow this table if too many data blocks are purged at one time; therefore, only those data blocks that can truly be prepurged should be done so. An example of explicit and implicit purging and prepurging follows:

```

XDMAP      ERR=2 $
MØD1      IP/A/S,Y,PX/S,Y,PY/S,Y,PB $
PURGE     X/PX / Y/PY $
MØD2      A/B,C,D/V,Y,PB/S,Y,PC $
PURGE     C/PC $
MØD3      B,C,D/E $
MØD4      E/X,Y,Z $
MØD5      X,Y,Z// $
END        $

```

Assume that module MØD1 sets  $PX < 0$ ,  $PY \geq 0$ , and  $PB = 0$ . Assume that B is not generated by MØD2 if  $PB = 0$ . Assume that MØD2 sets  $PC < 0$  but does not change PB.

The following table shows what happens when the above DMAP program is executed. Only modules being executed are shown in the table.

Module being executed	Input status and comments	Output status and comments
MØD1	IP - assumed generated by the input file processor	A - generated PX < 0, PY ≥ 0, PB = 0 and save for use in subsequent modules
PURGE	X,Y - not generated PX < 0, PY ≥ 0	X - purged (i.e., prepurged) Y - not generated
MØD2	A - generated; PB = 0	B - purged (i.e., implicitly); C,D - generated; PC < 0; PC is saved for subsequent use
PURGE	C - generated PC < 0	C - purged
MØD3	B,C - purged D - generated	E - generated
MØD4	E - generated	X - purged; Y - generated; Z - generated
MØD5	X - purged Y,Z - generated	Output to printer (assumed)
END		Normal termination of problem

The following is an example of unpurging:

```

XDMAP      ERR=2 $
FILE      X=SAVE/Y=SAVE $
FILE      Z=APPEND $
MØD1      IP/A $
DMAP loop { LABEL      L1 $
           CØND      L2,NPX $
           PURGE     X/NPX $
MØD2      A/X,Y/S,Y,PX=0/S,N,NPX=0 $
           PURGE     X/PX $
           LABEL     L2 $
MØD3      X,Y/Z $
           REPT      L1,2,$
MØD4      Z// $
           END       $

```

Assume that MØD2 sets  $PX < 0$  and  $NPX \geq 0$  the first time it is executed.  
Assume that MØD2 sets  $PX \geq 0$  and  $NPX < 0$  the second time it is executed.

The following table shows what happens when this DMAP program is executed. Only modules being executed are shown in the table.

Module being executed	Input status and comments	Output status and comments
MØD1	IP - assumed generated by input file processor	A - generated
CØND1	NPX = 0	Jump not executed
PURGE	X - not generated	X - not generated (i.e., no action taken)
MØD2	A - generated	X,Y - generated; PX < 0 and NPX ≥ 0 are saved for subsequent use
PURGE	X - generated; PX < 0	X - purged
MØD3	X - purged; Y - generated	Z - generated
REPT	Loop count = 2	Transfer to location L1; loop count = 1
CØND	NPX ≥ 0	Jump not executed
PURGE	X - purged; NPX ≥ 0	X - not generated (i.e., unpurged)
MØD2	A - generated	X - generated; Y - generated (note old data for Y are lost because Y not appended); PX ≥ 0 and NPX < 0 are saved for subsequent use
PURGE	X - generated; PX ≥ 0	X - generated (i.e., no action taken)
MØD3	X,Y - generated	Z - generated (note new data appended to old because Z declared appended)
REPT	Loop count = 1	Transfer to location L1; loop count = 0
CØND	NPX < 0	Transfer to location L2
MØD3	X,Y - generated	Z - generated (i.e., appended)
REPT	Loop count = 0	Fall through to next instruction
MØD4	Z - generated	Output to printer (assumed)
END		Normal termination of problem

## 5. DIRECT MATRIX ABSTRACTION MODULES

### 5.1 Index of DMAP Modules Descriptions

Descriptions of all modules are contained herein and are arranged alphabetically by category, as indicated by the following lists:

#### Executive Operation Modules

COND	JUMP
END	LABEL
EQUIV	PURGE
EXIT	REPT
FILE	XDMAP

#### Structurally Oriented Modules

DPD	ØFP	SCE1
DSMG1	PLØT	SDR1
EMA	PLTSET	SDR2
EMG	PRTMSG	SMP1
GP1	RBMG1	SMP2
GP2	RBMG2	SSG1
GP3	RBMG3	SSG2
GP4	RBMG4	SSG3
GPSP	READ	TA1

#### Matrix Operation Modules

ADD	MPYAD
DECØMP	PARTN
FBS	SØLVE
MERGE	TRNSP

#### Utility Modules

CØPY	PARAML
DIAGONAL	PARAMR
INPUTT2	PRTPARM
MATPRN	SCALAR
ØUTPUT2	SETVAL
PARAM	SWITCH

In the examples that accompany each description, the following notation is used:

Upper-case letters and special symbols in the DMAP calling sequence must be punched as shown except for data block names, parameter names, and label names that are symbolic

Lower-case letters represent constants whose permissible values are indicated in the descriptive text



Due to the many possible forms which may be used when writing parameters, a variety of arbitrarily selected forms is used in the examples. This situation does not imply that the form used in any example is required or that it is the only acceptable form allowed.

The form, type, and precision of the terms are used in many functional module descriptions. By form, one of the following is meant:

Form	Meaning
1	Square
2	Rectangular
6	Symmetric

By type, one of the following is meant:

Type	Meaning
1	Real, single precision
2	Real, double precision
3	Complex, single precision
4	Complex, double precision

By precision, one of the following is meant:

Precision indicator	Meaning
1	Single-precision numbers
2	Double-precision numbers

## 5.2 Executive Operation Modules

Module	Basic function	Page
COND	Conditional forward jump	131
END	Always last in DMAP; terminates DMAP execution	132
EQUIV	Assigns another name to a data block	133
EXIT	Conditional DMAP termination	134
FILE	Defines special data block characteristics to DMAP compiler	135
JUMP	Unconditional forward jump	136
LABEL	Defines DMAP location	137

Module	Basic function	Page
PURGE	Conditional data block elimination	138
REPT	Repeats a series of DMAP instructions	.139
XDMAP	Controls the DMAP compiler options	141

All modules classified as executive operation modules are individually described in this section. Additional discussions concerning the interaction of the executive modules with themselves and with the NASTRAN executive system are contained in sections 4.9 to 4.13.

I. NAME: ~~C~~OND (Conditional transfer)

II. PURPOSE: To alter the normal order of execution of DMAP modules by conditionally transferring program control to a specified location in the DMAP program.

III. DMAP CALLING SEQUENCE:

~~C~~OND n,V \$

where

n BCD label name specifying the location where control is to be transferred (see LABEL instruction)

V BCD name of a variable parameter whose value indicates whether or not to execute the transfer. If  $V < 0$ , the transfer is executed.

IV. EXAMPLE:

BEGIN \$

.  
. .  
.

~~C~~OND L1,K \$

~~M~~ODULE1 A/B/V,Y,P1 \$

.  
. .  
.

LABEL L1 \$

~~M~~ODULEN X/Y \$

.  
. .  
.

END \$

If  $K \geq 0$ , ~~M~~ODULE1 is executed. If  $K < 0$ , control is transferred to the label L1 and ~~M~~ODULEN is executed.

V. REMARKS:

1. Only forward transfers are allowed. See the REPT instruction for backward transfers.

I. NAME: END (End DMAP program)

II. PURPOSE: Denotes the end of a DMAP program.

III. DMAP CALLING SEQUENCE:

END \$

IV. REMARKS:

1. The END instruction also acts as an implied EXIT instruction.

I. NAME: EQUIV (Data block name equivalence)

II. PURPOSE: To attach one or more equivalent (alias) data block names to an existing data block so that the data block can be referenced by several equivalent names.

III. DMAP CALLING SEQUENCE:

EQUIV DBN1A,DBN2A,DBN3A / PARMA / DBN1B,DBN2B / PARMB \$

The number of data block names (DBNij) prior to each parameter (PARMj) and the number of such groups in a particular calling sequence are variable.

IV. INPUT DATA BLOCKS:

DBN1A,DBN2A, etc. Any data block names that appear within the DMAP sequence. The first data block name in each group (DBN1A and DBN1B in the previous examples) is known as the primary data block and the second, etc., data block names become equivalent to the primary, depending on the associated parameter value. These equivalenced data blocks are known as secondary data blocks.

V. OUTPUT DATA BLOCKS: None specified or permitted.

VI. PARAMETERS:

PARMA, etc. One is required for each set of data block names.

VII. METHOD: The data block names in each group are made equivalent if the value of the associated parameter is less than 0. If a number of data blocks are already equivalenced and the parameter value is  $\geq 0$ , the equivalence is broken and the data block names again become unique. If the data blocks are not equivalenced and the parameter value is  $\geq 0$ , no action is taken.

VIII. REMARKS:

1. An EQUIV statement may appear at any time as long as the primary data block name has been previously defined.
2. If an equivalence is to be performed at all times (i.e., the parameter value is always negative), it is not necessary to specify a parameter name. For example,

EQUIV DB1,DB2 // DB3,DB4 \$

I. NAME: EXIT (Terminate DMAP program)

II. PURPOSE: To conditionally terminate the execution of the DMAP program.

III. DMAP CALLING SEQUENCE:

EXIT c \$ where c is an integer constant which specifies the number of times the instruction is to be ignored before terminating the program. If c = 0, the calling sequence may be shortened to EXIT \$.

IV. EXAMPLE:

```
      BEGIN $
      .
      .
      .
      LABEL L1 $
      MODULE A/B/V,Y,P1 $
      .
      .
      .
      EXIT 3 $
      REPT L1,3 $
      .
      .
      .
      END $
```

DMAP  
loop

V. REMARKS:

1. The EXIT instruction is executed the third time the loop is repeated (i.e., the instructions within the loop will be executed four times).
2. EXIT may appear anywhere within the DMAP sequence.

I. NAME: FILE (File allocation aide)

II. PURPOSE: To inform the file allocator of any special characteristics of a data block.

III. DMAP CALLING SEQUENCE:

FILE A=a1,a2...a $\alpha$  / B=b1,b2...b $\beta$  / .... / Z=z1,z2...z $\omega$  \$

where

A,B...Z names of data blocks possessing special characteristics

a1...a $\alpha$ ,b1...b $\beta$ ,...,z1...z $\omega$  special characteristics from the following list

The allowable special characteristics are:

SAVE Indicates data block is to be saved for possible looping in DMAP program.

APPEND Output data blocks which are generated within a DMAP loop are rewritten during each pass through the loop unless the data block is declared APPEND in a FILE statement. The APPEND declaration allows a module to add information to a data block on successive passes through a DMAP loop.

TAPE Indicates that data block is to be written on a physical tape if a physical tape is available.

Data blocks created by the NASTRAN preface may not appear in FILE declarations.

Symbolic DMAP sequences which explain the use of the FILE instruction are given in section 4.11.

FILE is a nonexecutable DMAP instruction which is used only by the DMAP compiler for information purposes.

A data block name may appear only once in all FILE statements; otherwise, the first appearance will determine all special characteristics applied to the data block.

I. NAME: JUMP (Unconditional transfer)

II. PURPOSE: To alter the normal order of execution of DMAP modules by unconditionally transferring program control to a specified location in the DMAP program. The normal order of execution of DMAP modules is the order of occurrence of the modules as DMAP instructions in the DMAP program.

III. DMAP CALLING SEQUENCE:

JUMP n \$ where n is a BCD name appearing on a LABEL instruction which specifies where control is to be transferred.

IV. REMARKS:

1. Jumps must be forward in the DMAP sequence. See the REPT instruction for backward jumps.



I. NAME: LABEL (DMAP location)

II. PURPOSE: To label a location in the DMAP program so that the location may be referenced by the DMAP instructions JUMP, COND, and REPT.

III. DMAP CALLING SEQUENCE:

LABEL n \$ where n is a BCD name.

IV. REMARKS:

1. The LABEL instruction is inserted just ahead of the DMAP instruction to be executed when transfer of control is made to the label.
2. LABEL is a nonexecutable DMAP instruction which is used only by the DMAP compiler for information purposes.

I. NAME: PURGE (Explicit data block purge)

II. PURPOSE: To flag a data block so that it will not be assigned to a physical file.

III. DMAP CALLING SEQUENCE:

PURGE DBN1A,DBN2A,DBN3A / PARMA / DBN1B,DBN2B / PARMB \$

The number of data block names (DBN<sub>i;j</sub>) prior to each parameter (PARM<sub>j</sub>) and the number of groups of data block names and parameters in a particular calling sequence are variable.

IV. INPUT DATA BLOCKS:

DBN1A,DBN2A, etc. Any data block names that appear within the DMAP sequence

V. OUTPUT DATA BLOCKS: None specified or permitted

VI. PARAMETERS:

PARMA, etc. One is required for each group of data block names.

VII. METHOD: The data blocks in a group are purged if the value of the associated parameter is less than 0. If a data block is already purged and the parameter value is  $\geq 0$ , the purged data block is unpurged so that it may be subsequently reallocated. If the data block is not purged and the parameter value is  $\geq 0$ , no action is taken.

VIII. REMARKS:

1. If a purge is to be made at all times (i.e., the parameter value is always negative), it is not necessary to specify a parameter name. For example,

PURGE DB1,DB2,DB3,DB4 \$

I. NAME: REPT (Repeat)

II. PURPOSE: To repeat a group of DMAP instructions a specified number of times.

III. DMAP CALLING SEQUENCE:

REPT n,c \$ ØR REPT n,p \$

where

n BCD name appearing in a LABEL instruction which specifies the location of the beginning of a group of DMAP instructions to be repeated (see LABEL instruction)

c An integer constant hard coded into the DMAP program which specifies the number of times to repeat the instructions

p A variable parameter set by a previously executed module which specifies the number of times to repeat the instructions

IV. EXAMPLES:

BEGIN \$		BEGIN \$
.		.
.		.
.		.
LABEL L1 \$		LABEL L1 \$
<del>M</del> ODULE1 A/B/V,Y,P1 \$		<del>M</del> ODULE1 A/B/V,Y,P1 \$
.		.
.	OR	<del>M</del> ODULE1 B/C/V,Y,NLØØP \$
.		.
<del>M</del> ODULEN B/C/V,Y,P2 \$		<del>M</del> ODULEN C/D/V,Y,P2 \$
REPT L1,3 \$		REPT L1,NLØØP \$
.		.
.		.
.		.
END \$		END \$

V. REMARKS:

1. REPT is placed at the end of the group of instructions to be repeated.
2. When a variable number of loops are to be performed, as in the second example in the Examples section, the value of the variable at the first time the REPT instruction is encountered will determine the number of loops. This number will not be changed after the initial assignment.
3. A COND (conditional jump) instruction may be used to exit from the loop if desired.
4. In the first example, the instructions MØDULE1 to MØDULEN will be repeated three times (i.e., executed four times).

I. NAME: XDMAP (Execute DMAP program)

II. PURPOSE: To control the DMAP compiler options.

III. DMAP CALLING SEQUENCE:

XDMAP {  $\frac{\emptyset}{N\emptyset\emptyset}$  }, {  $\frac{ERR=0}{ERR=1}$  /  $\frac{ERR=1}{ERR=2}$  }, {  $\frac{LIST}{N\emptyset LIST}$  }, {  $\frac{\emptyset SCAR}{N\emptyset\emptyset SCR}$  }, {  $\frac{REF}{N\emptyset REF}$  }

where

$\emptyset$  Compiles and executes program (default)

$N\emptyset\emptyset$  Compiles only and terminates job

ERR Defines the error level at which suspension of execution will occur

ERR = 0 warning error level

ERR = 1 potentially fatal error level

ERR = 2 fatal error level

LIST Listing of the DMAP program will be printed

$N\emptyset LIST$  No listing (default)

DECK Deck of the DMAP program will be punched

$N\emptyset DECK$  Deck will not be punched (default)

$\emptyset SCAR$  Detailed listing of  $\emptyset SCAR$  (operation sequence control array); output of the DMAP compiler

$N\emptyset\emptyset SCR$  No  $\emptyset SCAR$  listing (default)

REF Cross-reference listing of the program will be printed

$N\emptyset REF$  No cross-reference (default)

IV. REMARKS:

1. The XDMAP instruction is nonexecutable and is used to control the previously mentioned options by the DMAP compiler.

### 5.3 Structurally Oriented Modules

Module	Basic function	Page
DPD	Dynamics pool distributor	144
DSMG1	Differential stiffness matrix generator	146
EMA	Element matrix assembler	148
EMG	Element generator	149
GP1	Geometry processor - phase 1	151
GP2	Geometry processor - phase 2	152
GP3	Geometry processor - phase 3	153
GP4	Geometry processor - phase 4	154
GPSP	Grid-point singularity processor	156
ØFP	Output file processor	157
PLØT	Structural plotter	158
PLTSET	Plot set definition processor	160
PRTMSG	Message writer	161
RBMG1	Rigid-body matrix generator - phase 1	162
RBMG2	Rigid-body matrix generator - phase 2	163
RBMG3	Rigid-body matrix generator - phase 3	164
RBMG4	Rigid-body matrix generator - phase 4	165
READ	Real eigenvalue analysis	166
SCE1	Single-point constraint eliminator	168
SDR1	Stress data recovery - phase 1	170
SDR2	Stress data recovery - phase 2	172

Module	Basic function	Page
SMP1	Structural matrix partitioner - phase 1	175
SMP2	Structural matrix partitioner - phase 2	176
SSG1	Static solution generator - phase 1	177
SSG2	Static solution generator - phase 2	178
SSG3	Static solution generator - phase 3	180
TA1	Table assembler	182

I. NAME: DPD (Dynamics pool distributor)

II. PURPOSE: DPD is the principal data processing module for dynamics problems. New tables are assembled to account for any extra points in the model and the additional displacement sets used in dynamics. Bulk data cards that control the solution of a dynamics problem are processed and assembled into various data blocks for convenience and efficiency in solving the dynamics problem.

III. DMAP CALLING SEQUENCE:

DPD DYNAMICS,GPL,SIL,USET/GPLD,SILD,USETD,TFPOOL,DLT,PSDL,FRL,NLFT,TRL,  
EED,EQDYN/V,N,LUSET/V,N,LUSETD/V,N,NOTFL/V,N,NOTDLT/V,N,NOTPSDL/V,N,  
NOTFRL/V,N,NOTNLFT/V,N,NOTTRL/V,N,NOTEED/C,N,OV,N,NOUE \$

IV. INPUT DATA BLOCKS:

DYNAMICS Collection of bulk data cards for dynamics problem  
GPL Grid-point list  
SIL Scalar index list  
USET Displacement set definitions table

DYNAMICS may be purged. Other input data blocks may not be purged.

V. OUTPUT DATA BLOCKS:

GPLD Grid-point list dynamics  
SILD Scalar index list dynamics  
USETD Displacement set definition table dynamics  
TFPOOL Transfer function pool  
DLT Dynamic loads table  
PSDL Power spectral density list  
FRL Frequency response list  
NLFT Nonlinear forcing table  
TRL Transient response list  
EED Eigenvalue extraction data  
EQDYN Equivalence between external and internal numbers; dynamics



GPLD,SILD,USETD, and EQDYN may not be purged. All other data blocks may be purged.

VI. PARAMETERS:

LUSET	Input; integer; no default. Degrees of freedom in the g displacement set.
LUSETD	Output; integer; no default. Degrees of freedom in the p displacement set.
NØTFL	Output; integer; no default. Number of transfer function sets in the bulk data; -1 if no sets are defined.
NØDLT	Output; integer; no default. +1 if dynamics load data are present in the bulk data (i.e., DLT is created); -1 otherwise.
NØPSDL	Output; integer; no default. +1 if the PSDL is created; -1 otherwise.
NØFRL	Output; integer; no default. +1 if the FRL is created; -1 otherwise.
NØNLFT	Output; integer; no default. +1 if the NLFT is created; -1 otherwise.
NØTRL	Output; integer; no default. +1 if the TRL is created; -1 otherwise.
NØEED	Output; integer; no default. +1 if the EED is created; -1 otherwise.
NØUE	Output; integer; no default. Number of extra points in the model; -1 if there are no extra points.

- I. NAME: DSMG1 (Differential stiffness matrix generator)
- II. PURPOSE: To generate the differential stiffness matrix  $\begin{bmatrix} d \\ K_{gg} \end{bmatrix}$  which is used in buckling analysis.
- III. DMAP CALLING SEQUENCE:  
 DSMG1 CASECC,SIL,EDT,UGV,CSTM,MPT,ECPT,GPCT,DIT/KDNN/V,N,DSCØSET \$

IV. INPUT DATA BLOCKS:

CASECC	Case control data table
SIL	Scalar index list
EDT	Element deformation table
UGV	Initial approximation to the displacement vector; g set
CSTM	Coordinate system transformation matrices
MPT	Material properties table
ECPT	Element connection and properties table
GPCT	Grid-point connection table
DIT	Direct input tables

A fatal error exists if CASECC is purged.

A fatal error exists if an element deformation set is requested by the user in case control and EDT is purged.

A fatal error exists if UGV is purged.

CSTM can be purged. However, if some grid point of the model is not in basic coordinates and the CSTM is purged, a fatal error occurs.

If the MPT is purged and some element references a material property, a fatal error occurs.

A fatal error occurs if the ECPT is purged.

A fatal error occurs if the GPCT is purged.

V. OUTPUT DATA BLOCKS:

KDNN Partition of differential stiffness matrix; g set  
KDNN cannot be prepurged. A fatal error occurs if it is.

VI. PARAMETERS:

DSCØSET Output; integer; no default value

- I. NAME: EMA (Element matrix assembler)
- II. PURPOSE: To superimpose matrices corresponding to elements into a structural matrix corresponding to all degrees of freedom at all grid points.
- III. DMAP CALLING SEQUENCE:
- EMA GPECT,XEMD,XMAT / XGG,GPST / C,N,NØK4 / C,N,WTMASS \$
- IV. INPUT DATA BLOØCKS:
- GPECT Grid-point element connection table
- XEMD X-matrix element matrix dictionaries (X = K, M, B, or KD)
- XMAT Element matrices
- V. OUTPUT DATA BLOØCKS:
- XGG Structural matrix (X = K, M, B, KD, or K2)
- GPST Grid-point singularity table
- VI. PARAMETERS:
- NØK4 Input; integer; no default. Flag which specifies whether damping factor is to be used in assembling matrix (-1 ignores factor).
- WTMSS Input; floating point; Default = 1.0. Constant by which all element matrix terms are multiplied.

I. NAME: EMG (Element generator)

II. PURPOSE: This module will compute and output stiffness and mass matrices for individual elements. The module also produces auxiliary description and location data (i.e., dictionary information) with respect to the above element stiffness and mass matrices. This module does not assemble these element matrices with regard to some element linkage specification. The assembly process is performed by module EMA.

III. DMAP CALLING SEQUENCE:

EMG EST,CSTM,MPT,DIT,GEOM2,/KELM,KDICT,MELM,MDICT,BELM,BDICT/V,N,  
NOK/V,N,NOM/V,N,NOB/V,N,NOK4GG/V,N,NOKDGG/V,N,CONMASS/ \$

IV. INPUT DATA BLOCKS:

EST Element summary table  
CSTM Coordinate system transformation matrices  
MPT Material properties table  
DIT Direct input table  
GEOM2 Geometry table 2

The CSTM may be purged. The MPT may be purged only if elements which do not reference any material data are used.

V. OUTPUT DATA BLOCKS:

KELM Element stiffness matrix partitions  
KDICT Dictionary table for element stiffness matrix partitions  
MELM Element mass matrix partitions  
MDICT Dictionary table for element mass matrix partitions  
BELM Element damping matrix partitions  
BDICT Dictionary table for element damping matrix partitions

If either of a matrix-dictionary data block pair is purged, that particular data block pair will not be formed.

## VI. PARAMETERS:

- NØK** Input; integer; no default. A value  $\leq 0$  implies that they do not form stiffness matrix and dictionary data blocks. A value  $> 0$  implies that they form stiffness matrix and dictionary data blocks.
- NØM** Same as NØK but for mass matrices.
- NØB** Same as NØK.
- NØK4GG** Output; Default = -1.
- NØKDGG** Currently not used.
- ØNMASS** Integer; input; no default. A value  $\leq 0$  implies that they do not form consistent mass matrices. A value  $> 0$  implies that they form consistent mass matrices.

I. NAME: GP1. (Geometry processor - phase 1)

II. PURPOSE: GP1 performs basic geometry processing for the model. A list of all grid and scalar points is assembled and placed in internal order. Coordinate system transformation matrices are computed, and all grid points are transformed to the basic coordinate system.

III. DMAP CALLING SEQUENCE:

```
GP1 GEOM1,GEOM2,X/GPL,EQEXIN,GPDT,CSTM,BGPDT,SIL/V,N,LUSET/V,  
N,NOCSTM/V,N,NOPDPT $
```

IV. INPUT DATA BLOCKS:

GEOM1 Grid point; coordinate system; sequence data

GEOM2 Element connection data

X Dummy data block

V. OUTPUT DATA BLOCKS:

GPL Grid-point list

EQEXIN Equivalence between external grid or scalar numbers and internal numbers

GPDT Grid-point definition table

CSTM Coordinate system transformation matrices

BGPDT Basic grid-point definition table

SIL Scalar index list

No output data block may be purged.

VI. PARAMETERS:

LUSET Output; integer; no default. Total degrees of freedom in the g displacement set.

NOCSTM Output; integer; no default. Number of coordinate systems defined in the bulk data deck; -1 if no coordinate systems defined.

NOPDPT Output; integer; no default. -1 if no grid or scalar points defined in bulk data deck; +1 otherwise.

I. NAME: GP2 (Geometry processor - phase 2)

II. PURPOSE: GP2 processes element connection data and converts external point numbers to internal numbers.

III. DMAP CALLING SEQUENCE:

GP2 GEOM2,EQEXIN/ECT \$

IV. INPUT DATA BLOCKS:

GEOM2 Element connection data

EQEXIN Equivalence between external grid or scalar numbers and internal numbers

EQEXIN may not be purged.

V. OUTPUT DATA BLOCKS:

ECT Element connection table

ECT may not be purged.

VI. PARAMETERS: None



I. NAME: GP3 (Geometry processor - phase 3)

II. PURPOSE: GP3 processes static loads data. Static loads data are collected by set, and external numbers are converted to internal numbers.

III. DMAP CALLING SEQUENCE:

GP3 ~~GEOM3~~,EQEXIN,~~GEOM2~~/SLT/V,N,~~NLOAD~~/V,N,~~NGRAV~~/V,N,~~NOTEMP~~ \$

IV. INPUT DATA BLOCKS:

~~GEOM3~~ Static loads and temperature data

EQEXIN Equivalence between external grid and scalar numbers and internal numbers

~~GEOM2~~ Element connection data

EQEXIN may not be purged.

V. OUTPUT DATA BLOCKS:

SLT Static loads table

VI. PARAMETERS:

~~NLOAD~~ Output; integer; no default. -1 if no static loads (i.e., SLT is not created); +1 otherwise.

~~NGRAV~~ Output; integer; -1.

~~NOTEMP~~ Output; integer; -1.

I. NAME: GP4 (Geometry processor - phase 4)

II. PURPOSE: GP4 assembles the various displacement sets and builds the displacement set definition table (USET). Additionally, for statics problems, GP4 analyzes subcases based on single-point constraint sets and set parameters to control execution of the DMAP sequence.

III. DMAP CALLING SEQUENCE:

```
GP4 CASECC,GEOM4,EQEXIN,SIL,GPDT,BGPDT,CSTM / YS,USET /V,N,LUSET /
V,N,MPCF1 / V,N,MPCF2 / V,N,SINGLE / V,N,OMIT / V,N,REACT /
V,N,NSKIP / V,N,REPEAT / V,N,NOSSET / V,N,NOL / V,N,NOA /
C,N,SSID $
```

IV. INPUT DATA BLOCKS:

CASECC	Case control data table
GEOM4	Displacement set definition
EQEXIN	Equivalence between external grid and internal numbers
SIL	Scalar index list
GPDT	Grid-point definition table
BGPDT	Basic grid-point definition table
CSTM	Coordinate system transformation matrix table

Only GEOM4 and CSTM may be purged.

V. OUTPUT DATA BLOCKS:

YS	Constrained displacement vector(s) set
USET	Displacement set definition table

YS may be purged.

VI. PARAMETERS:

LUSET	Input; integer; no default. Degrees of freedom in the g displacement set.
MPCF1	Output; integer; -1.
MPCF2	Output; integer; -1.

**SINGLE** Output; integer; no default. +1 if the current subcase contains single-point constraints; -1 otherwise.

**ØMIT** Output; integer; no default. +1 if the model contains omitted coordinates; -1 otherwise.

**REACT** Output; integer; no default. +1 if the model contains supports; -1 otherwise.

**NSKIP** Input and output; integer; Default = 0. Number of records to skip to reach the first record in the case control data block for the next subcase. (NSKIP = 0 for the first subcase.)

**REPEAT** Output; integer; no default. -1 if the current subcase is the last subcase in the problem; +1 otherwise.

**NOSET** Output; integer; no default. -1 if MPCF1 = -1, SINGLE = -1, ØMIT = -1, and REACT = -1; +1 otherwise.

**NOFL** Output; integer; Default = +1. -1 if all degrees of freedom in the model belong to dependent displacement sets (i.e., no degree of freedom belongs to an independent set); +1 otherwise.

**NOA** Output; integer; Default = +1. -1 if MPCF1 = -1, SINGLE = -1, and ØMIT = -1; +1 otherwise.

**SSID** Input; integer; Default = 0. Reserved for future use.

- I. NAME: GPSP (Grid-point singularity processor)
- II. PURPOSE: The GPST data block contains data on possible stiffness matrix singularities. These singularities may be removed through the application of single-point constraints. The GPSP module checks each singularity against the list of constraints, and if the singularity is not removed, writes data for warning the user.
- III. DMAP CALLING SEQUENCE:
- GPSP GPL,GPST,USET,SIL / ØGPST / V,N,ØGPS \$
- IV. INPUT DATA BLOCKS:
- GPL Grid-point list
- GPST Grid-point singularity table
- USET Displacement set definitions table
- SIL Scalar index list
- No input data block can be purged.
- V. OUTPUT DATA BLOCKS:
- ØGPST Unremoved grid-point singularities. This data block will be processed by the ØFP (output file processor) module.
- VI. PARAMETERS:
- ØGPS Output; integer; Default = 1. If ØGPS < 0, then ØGPST is empty.

I. NAME: ØFP (Output file processor)

II. PURPOSE: ØFP outputs to the system output file, in user-oriented, self-explanatory formats, data blocks prepared for output by other functional modules.

III. DMAP CALLING SEQUENCE:

ØFP DB1,DB2,DB3,DB4,DB5,DB6//V,N,CARDNØ \$

IV. INPUT DATA BLOCKS:

One to six input data blocks are output in the order desired. Any or all input data blocks may be purged.

V. OUTPUT DATA BLOCKS: None

VI. PARAMETERS:

CARDNØ Input and output; integer; Default = 0. CARDNØ is incremented by one and punched in columns 73 to 80 for each card punched by ØFP.

I. NAME: `PLØT` (Structural plotter)

II. PURPOSE: To draw structural shapes on a variety of different plotters.

III. DMAP CALLING SEQUENCE:

```
PLØT PLTPAR,GPSETS,ELSETS,CASECC,BGPDT,EQEXIN,SIL,PLTDSP1,PLTDSP2,  
ECPT,ØES1/PLØTX/V,N,NGP/V,N,LSIL/V,N,NPSET/V,N,PLTFLG/V,N,  
PLTNUM $
```

IV. INPUT DATA BLOCKS:

<code>PLTPAR</code>	Plot parameters and plot control table
<code>GPSETS</code>	Grid-point sets related to the element plot sets
<code>ELSETS</code>	Element plot set connection tables
<code>CASECC</code>	Case control data table
<code>BGPDT</code>	Basic grid-point definition table
<code>EQEXIN</code>	Equivalence between external grid or scalar numbers and internal numbers
<code>SIL</code>	Scalar index list
<code>PLTDSP1</code>	Translational deformation (statics)
<code>PLTDSP2</code>	Translational deformations (dynamics)
<code>ECPT</code>	Grid-point element connection table
<code>ØES1</code>	Output element stress requests

Only `SIL`, `PLTDSP1`, and `PLTDSP2` may be purged. If this is the case, only undeformed shapes may be drawn.

If either `PLTDSP1` or `PLTDSP2` is purged, that type of deformed shape will not be drawn.

If either `PLTDSP1` or `PLTDSP2` is not purged, `SIL` may not be purged.

V. OUTPUT DATA BLOCKS:

`PLØTX` User messages

`PLØTX` may not be purged.

VI. PARAMETERS:

NGP            Integer; input; no default value.    Number of grid points.

LSIL           Integer; input; no default value.    Last scalar index value.

NPSET          Integer; input; no default value.    Number of element plot sets.

PLTFLG        Integer; input/output;    Default value = 1.    Displacement plot  
flag.    1 if undeformed shapes have not yet been drawn; -1 if  
undeformed shapes have been drawn.

PLTNUM        Integer; input/output;    Default value = 0.    Plot number.

I. NAME: PLTSET (Plot set definition processor)

II. PURPOSE: To generate the structural element sets to be used by the structural plotter (functional module PLØT).

III. DMAP CALLING SEQUENCE:

PLTSET PCDB,EQEXIN,ECT/PLTSETX,PLTPAR,GPSETS,ELSETS/V,N,NGP/V,N,NPSET \$

IV. INPUT DATA BLOØCKS:

PCDB Plot control data block for the structure plotter

EQEXIN Equivalence between external grid or scalar numbers and internal numbers

ECT Element connection table

If PCDB is purged, nothing is done in this module. However, if PCDB is not purged, neither EQEXIN nor ECT may be purged.

V. OUTPUT DATA BLOØCKS:

PLTSETX User error messages related to the definition of element plot sets for the structure plotter

PLTPAR Plot parameters and plot control table

GPSETS Grid-point sets related to the element plot sets

ELSETS Element plot set connection tables

None of these data blocks may be prepurged unless PCDB is also purged.

VI. PARAMETERS:

NGP Output; integer; no default. Total number of grid points.

NPSET Output; integer; Default value = -1. Number of element plot sets (set to -1 if none).



- I. NAME: PRTMSG (Message writer)
- II. PURPOSE: To process a data block of user-oriented messages.
- III. DMAP CALLING SEQUENCE:  
PRTMSG MSG// \$
- IV. INPUT DATA BLOCKS:  
MSG Messages to be printed (if purged, nothing is done)
- V. OUTPUT DATA BLOCKS: None
- VI. PARAMETERS: None

I. NAME: RBMG1 (Rigid-body matrix generator - phase 1)

II. PURPOSE: RBMG1 partitions  $[K_{aa}]$  into  $[K_{\ell\ell}]$ ,  $[K_{\ell r}]$ , and  $[K_{rr}]$ .

If  $[M_{aa}]$  is not purged, it is partitioned similarly.

$$[K_{aa}] = \begin{bmatrix} K_{\ell\ell} & | & K_{\ell r} \\ \hline K_{r\ell} & | & K_{rr} \end{bmatrix}$$

III. DMAP CALLING SEQUENCE:

RBMG1 USET, KAA, MAA/KLL, KLR, KRR, MLL, MLR, MRR \$

IV. INPUT DATA BLOCKS:

USET Displacement set definitions table

KAA Partition of stiffness matrix; a set

MAA Partition of mass matrix; a set

USET may not be purged.

V. OUTPUT DATA BLOCKS:

KLL Partition of stiffness matrix;  $\ell$  set

KLR Partition of stiffness matrix

KRR Partition of stiffness matrix;  $r$  set

MLL Partition of mass matrix;  $\ell$  set

MLR Partition of mass matrix

MRR Partition of mass matrix;  $r$  set

Output data blocks may be purged only if the corresponding input data block is purged.

VI. PARAMETERS: None

I. NAME: RBMG2 (Rigid-body matrix generator - phase 2)

II. PURPOSE: RBMG2 decomposes  $[K_{ll}]$  into its triangular factors  $[L_{ll}]$  and  $[U_{ll}]$ .

III. DMAP CALLING SEQUENCE:

RBMG2 KLL/LLL/V,N,PPOWER/V,N,DET \$

IV. INPUT DATA BLOCKS:

KLL Partition of stiffness matrix;  $l$  set

KLL may not be purged.

V. OUTPUT DATA BLOCKS:

LLL Lower triangular factor of KLL;  $l$  set

LLL may not be purged.

VI. PARAMETERS:

PPOWER Output; integer; Default = 1. Power of 10 in the determinant of KLL.

DET Output; real; Default = 1.0. Magnitude of determinant of KLL (i.e.,  $\det [K_{ll}] = \text{DET} * 10^{\text{PPOWER}}$ ).

I. NAME: RBMG3 (Rigid-body matrix generator - phase 3)

II. PURPOSE: RBMG3 solves for the rigid-body transformation matrix  $[D]$  from the equation

$$[K_{ll}] [D] = -[K_{lr}] \quad (1)$$

The rigid-body error ratio  $\epsilon$  is computed from

$$\epsilon = \frac{\| [K_{rr}] + [K_{lr}]^T [D] \|}{\| [K_{rr}] \|} \quad (2)$$

The absolute value  $\| \|$  is the square root of the sum of the squares (not a determinant).

III. DMAP CALLING SEQUENCE:

RBMG3 LLL,KLR,KRR/DM \$

IV. INPUT DATA BLOCKS:

LLL Lower triangular factor of KLL;  $l$  set

KLR Partition of stiffness matrix

KRR Partition of stiffness matrix;  $r$  set

Input data blocks may not be purged.

V. OUTPUT DATA BLOCKS:

DM Rigid-body transformation matrix

The DM data block corresponds to the matrix  $[D]$  and may not be purged.

VI. PARAMETERS: None

I. NAME: RBMG4 (Rigid-body matrix generator - phase 4)

II. PURPOSE: RBMG4 computes the rigid-body mass matrix  $[m_r]$  from the matrix equation

$$[m_r] = [M_{rr}] + [D]^T [M_{\ell r}] + [M_{\ell r}]^T [D] + [D]^T [M_{\ell \ell}] [D] \quad (1)$$

III. DMAP CALLING SEQUENCE:

RBMG4 DM,MLL,MLR,MRR/MR \$

IV. INPUT DATA BLOCKS:

DM Rigid-body transformation matrix  
MLL Partition of mass matrix;  $\ell$  set  
MLR Partition of mass matrix  
MRR Partition of mass matrix;  $r$  set

No input data block may be purged. The DM data block corresponds to the matrix  $[D]$  in equation (1).

V. OUTPUT DATA BLOCKS:

MR Rigid-body mass matrix;  $r$  set

VI. PARAMETERS: None

I. NAME: READ (Real eigenvalue analysis)

II. PURPOSE: To solve the equation

$$[K] - \lambda [M] \{u\} = 0$$

for eigenvalues  $\lambda$  and their associated eigenvectors.

III. DMAP CALLING SEQUENCE:

READ KAA, { KDAAM  
MAA }, MR, DM, EED, USET, CASECC/LAMA, PHIA, MI, ØEIGS/V, N,  
FORMAT/V, N, NEIGVS/V, N, NSKIP \$

IV. INPUT DATA BLOCKS:

KAA Partition of stiffness matrix; a set

KDAAM Negative of partition of differential stiffness matrix  $[K_{aa}^d]$ ;  
a set

MAA Partition of mass matrix; a set

MR Rigid-body mass matrix; r set

DM Rigid-body transformation matrix

EED Eigenvalue extraction data

USET Displacement set definitions table

CASECC Case control data table

KAA must be present.

MR may or may not be present.

DM and USET must be present if MR is present.

EED and CASECC must be present.

In buckling analysis, MAA = -KDAAM.

## V. OUTPUT DATA BLOCKS:

LAMA            Real eigenvalue table

PHIA            Eigenvectors matrix that gives the eigenvectors in the a set

MI              Modal mass matrix

ØEIGS          Real eigenvalue summary table

LAMA and PHIA may also be input data blocks if the append mode is being used.

## VI. PARAMETERS:

FØRMAT        Input; BCD; no default. If FØRMAT ≠ MØDES, READ will solve a buckling problem (i.e.,  $[\lambda M - K]\{u\} = 0$ ) by using EIGB data cards where M is the negative of the differential stiffness matrix.

NEIGVS        Output; integer; no default. NEIGVS is the number of eigenvalues found. If none were found, NEIGVS = -1.

NSKIP         Input; integer; Default value = 1. The method used by READ is taken from the NSKIP record of CASECC.

I. NAME: SCE1 (Single-point constraint eliminator)

II. PURPOSE: To reduce the n set matrices to f set matrices by removing the single-point constraints.

III. DMAP CALLING SEQUENCE:

SCE1 USET,  $\left\{ \begin{array}{c} \text{KDNN} \\ \text{KNN} \end{array} \right\}$ , MNN, BNN, K4NN /  $\left\{ \begin{array}{c} \text{KDFF} \\ \text{KFF} \end{array} \right\}$ ,  $\left\{ \begin{array}{c} \text{KDFS} \\ \text{KFS} \end{array} \right\}$ ,  $\left\{ \begin{array}{c} \text{KDSS} \\ \text{KSS} \end{array} \right\}$ , MFF, BFF,  
K4FF / \$

IV. INPUT DATA BLOCKS:

USET Displacement set definitions table  
KDNN Partition of differential stiffness matrix; n set  
KNN Partition of stiffness matrix; n set  
MNN Partition of mass matrix; n set  
BNN Partition of damping matrix; n set  
K4NN Partition of the structural damping matrix; n set

USET cannot be purged.

KNN, MNN, BNN, and K4NN can be purged.

At least one degree of freedom must belong to the f and s sets.

V. OUTPUT DATA BLOCKS:

KDFF Partition of differential stiffness matrix after single-point constraints have been removed; f set  
KFF Partition of stiffness matrix after single-point constraints have been removed; f set  
KDFS Partition of differential stiffness matrix after single-point constraints have been removed  
KFS Partition of stiffness matrix after single-point constraints have been removed



KDSS      Partition of differential stiffness matrix after single-point constraints have been removed; s set

KSS        Partition of stiffness matrix after single-point constraints have been removed; s set

MPF        Partition of mass matrix after single-point constraints have been removed; f set

BFF        Partition of damping matrix after single-point constraints have been removed; f set

K4FF       Partition of structural damping matrix with single-point constraints removed; f set

VI. PARAMETERS: None

I. NAME: SDR1 (Stress data recovery - phase 1)

II. PURPOSE: The SDR1 module utilizes solution vectors to produce displacements, eigenvectors, velocities, accelerations, applied loads, and reaction loads. The vectors input to SDR1 are in the form of packed matrices, with each column a solution vector for a different subcase, either eigenvalue or load. The row position of each term in a vector corresponds to a degree of freedom in a unique displacement set. The relative position of the term must be converted to a relative position in the vector, which includes all displacement components in the system. The dependent components of the displacement vector are recovered and merged to produce a complete vector describing all degrees of freedom in the structural or dynamics model. In the static analysis, SDR1 collects solutions for each boundary condition onto a single file, which is convenient for the solution of symmetry problems.

III. DMAP CALLING SEQUENCE:

```
SDR1 USET,PG,PHIA,U00V,YS,G0,GM,PS,KFS,KSS,QR/PHIG,PGG,  
QG/V,N,APPEND/V,N,F0RMAT $
```

IV. INPUT DATA BLOCKS:

USET	Displacement set definition table
PG	Static load vector matrix giving static loads; g set
PHIA	Partition of the displacement vector matrix giving displacements in the l set (d set for transient response)
U <del>00</del> V	Partition of the displacement vector matrix giving displacements in the o set
YS	Constrained displacements; s set
G <del>0</del>	Structural matrix partitioning transformation matrix
GM	Rigid-body transformation matrix
PS	Partition of load vector matrix giving loads in s set
KFS	Partition of stiffness matrix after single-point constraints have been removed
KSS	Partition of stiffness matrix after single-point constraints have been removed; s set
QR	Determinant support forces matrix; r set

The first input block must always be present.

The second input block may or may not be present.

The third input block must always be present.

The fourth input block must be present unless the o set is null or FØRMAT = DYNAMICS (see section labeled "Parameters").

The fifth input block may or may not be present.

The sixth input block must be present unless the o set is null.

The seventh input block must be present unless the m set is null.

The eighth input block may or may not be present.

The ninth input block must be present unless the s set or the third output block is not present.

The tenth input block must be present unless the fifth input block is absent, the s set is null, or the third output block is not present.

The eleventh input block may or may not be present.

#### V. ØUTPUT DATA BLØCKS:

PHIG	Displacement vector matrix giving displacements in the g set
PGG	Static load vector appended to include all boundary conditions; g set
QG	Single-point constraint forces and determinate support forces matrix; g set

#### VI. PARAMETERS:

APPEND	Input; integer; no default
FØRMAT	Input; BCD; no default. Format indicates the problem type:
STATICS	Statics type problem
REIGEN	Real eigenvalue problem
DYNAMICS	Dynamic problem

I. NAME: SDR2 (Stress data recovery - phase 2)

II. PURPOSE: The SDR2 module processes the output requests for forces of single-point constraint, loads, point displacements, point velocities, point accelerations, element stresses, and element forces and formats the output data blocks with these final output results for direct outputting by the output file processor (ØFP) module or input to the SØRT2 processor (SDR3) module and then the XY-output modules (XYTRAN and XYPLØT).

III. DMAP CALLING SEQUENCE:

SDR2 CASECC,CSTM,MPT,DIT, {EQEXIN  
EQDYN},SIL,EDT,

BGPDT, {LAMA  
PGG  
TØL  
LAMA}, {QGI  
QG  
QBG}, {UGV  
UGV1  
PHIG},EST,YXCDB,PGG /

ØPG1,ØQGI, {ØPHIG  
ØUGV1},ØES1,ØEF1,PPHIG /

C,N, {STATICS  
REIGEN  
BLKØ  
BLK1} /V,N,NØSØRT2 \$

IV. INPUT DATA BLOCKS:

CASECC	Case control data table
CSTM	Coordinate system transformation matrices
MPT	Material property table
DIT	Direct input tables
EQEXIN	Equivalence between external grid or scalar numbers and internal numbers
EQDYN	Equivalence between external points and scalar index values
SIL	Scalar index list
EDT	Element deformation table
BGPDT	Basic grid-point definition table

PGG	Static load vector appended to include all boundary conditions
PGV1	Matrix of successive sums of incremental load vectors
TØL	Table of output times
LAMA	Real eigenvalue table
QG	Single-point constraint forces and determinant support forces matrix
QBG	Single-point forces of constraint matrix for differential stiffness; g set
QGI	Matrix of successive sums of incremental vectors of single-point constraint forces
UGV	Displacement vector matrix giving displacements in the g set
UGV1	Matrix of successive sums of incremental displacement vectors
PHIG	Eigenvector matrix giving eigenvectors
EST	Element summary table
XYCDB	XY case control data block

If the first input data block is purged, it is a fatal error. This data block is called case control in this module functional description.

The CSTM may be purged if no coordinate systems are referenced, or if stresses and/or forces are not requested.

The MPT may be purged if no stress or force requests are present.

The DIT may be purged if no stress or force requests are present, or if no temperature dependent materials are referenced.

The second record of EQEXIN must exist if a request exists for any loads, forces of single-point constraint, displacements, velocities, accelerations, or plots.

SIL may be purged if no stress or force requests exist or if there are no extra points and no plots. (The second record is used by SDR2.)

The EDT may be purged if there are no element requests for forces or stresses, or if there are no enforced element deformations in the problem.

The BGPDT may be purged if the problem is in basic coordinates and no element requests for stresses or forces exist. No plots will result however.

LAMA may not be purged in an eigenvalue problem.

If input data block 11 (QG) is purged, forces of single-point constraint requests are ignored.

If input data block 12 (UGV or UGV1, etc.) is purged, SDR2 will process only loads and forces of single-point constraint requests.

If the EST or ESTIL is purged, element stresses and force requests are ignored.

The XYCDB may be purged.

#### V. OUTPUT DATA BLOCKS:

ØPG1	Output load vector requests
ØQG1	Output forces of single-point constraint requests
ØUGV1	Output displacement vector requests
ØPHIG	Output eigenvector requests
ØES1	Output element stress requests
ØEF1	Output element force requests
PPHIG	Translation components of the displacement vector rotated to basic coordinates

Output data blocks purged will result in output requests to those data blocks not being processed.

#### VI. PARAMETERS:

Name		Approach code
STATICS	BCD constant indicating a statics solution	1
REIGEN	BCD constant indicating a real eigenvalue solution	2
	BCD constant indicating the statics phase of a buckling solution	7
	BCD constant indicating the final phase of a buckling solution	8
ØSØRT2	Integer; output; set to 0 if there are no SØRT2 requests or requirements; set to 1 otherwise	

The approach code is placed in the output data blocks.

I. NAME: SMP1 (Structural matrix partitioner - phase 1)

II. PURPOSE: SMP1 partitions  $[K_{ff}]$  into  $[\bar{K}_{aa}]$ ,  $[K_{Oa}]$ , and  $[K_{Oo}]$ .

The matrix equation  $[K_{Oo}][G_o] = -[K_o]$  is solved for  $[G_o]$ .  $[K_{ff}]$  is then reduced by the matrix equation  $[K_{aa}] = [\bar{K}_{aa}] + [K_{Oa}]^T [G_o]$ .

If  $[M_{ff}]$  is not purged, it is reduced by the equation

$$[M_{aa}] = [\bar{M}_{aa}] + [G_o]^T [M_{Oa}] + [M_{Oa}]^T [G_o] + [G_o]^T [M_{Oo}] [G_o].$$

III. DMAP CALLING SEQUENCE:

SMP1 USET,KFF,MFF/GØ, KAA,KØØ,LØØ,MAA,MØØ,MØA / \$

IV. INPUT DATA BLOCKS:

USET Displacement set definitions table  
KFF Partition of stiffness matrix; f set  
MFF Partition of mass matrix; f set  
MFF may be purged.

V. OUTPUT DATA BLOCKS:

GØ Structural matrix partitioning transformation matrix  
KAA Partition of stiffness matrix; a set  
KØØ Partition of stiffness matrix; o set  
LØØ Lower triangular factor of KØØB; o set  
MAA Partition of mass matrix; a set  
MØØ Partition of mass matrix; a set  
MØA Partition of mass matrix

UØØ and LØØ are not standard-form matrices. Their format is compatible only for input to subroutine FBS.

MAA, MØØ, or MØA may be purged only if MFF is purged.

VI. PARAMETERS: None

I. NAME: SMP2 (Structural matrix partitioner - phase 2)

II. PURPOSE: To perform the following matrix operations:

$$\begin{bmatrix} \bar{d} \\ K_{fff} \end{bmatrix} \Rightarrow \begin{bmatrix} \bar{d} & d \\ \bar{K}_{aa} & K_{ao} \\ \hline [d]^T & d \\ K_{ao} & K_{oo} \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} \bar{d} \\ K_{aa} \end{bmatrix} = \begin{bmatrix} \bar{d} \\ \bar{K}_{aa} \end{bmatrix} + \begin{bmatrix} \bar{d} \\ K_{ao} \end{bmatrix} [G_o] + \left( \begin{bmatrix} \bar{d} \\ K_{ao} \end{bmatrix} [G_o] \right)^T + [G_o]^T \begin{bmatrix} \bar{d} \\ K_{oo} \end{bmatrix} [G_o] \quad (2)$$

III. DMAP CALLING SEQUENCE:

SMP2 USET, GØ, KDFF/KDAA / \$

IV. INPUT DATA BLOCKS:

USET Displacement set definitions table

GØ Structural matrix partitioning transformation matrix

KDFF Partition of differential stiffness matrix; f set

V. OUTPUT DATA BLOCKS:

KDAA Partition of differential stiffness matrix; a set

VI. PARAMETERS: None



I. NAME: SSG1 (Static solution generator - phase 1)

II. PURPOSE: To compute the static loads, thermal loads, and enforced deformation loads selected by the user.

III. DMAP CALLING SEQUENCE:

SSG1 SLT,BGPDT,CSTM,SIL,EST,MPT,EDT,MGG,CASECC/PG/V,N,LUSET/V,N,NSKIP \$

IV. INPUT DATA BLOCKS:

SLT Static loads table  
BGPDT Basic grid-point definition table  
CSTM Coordinate system transformation matrices  
SIL Scalar index list  
EST Element summary table  
MPT Material property table  
EDT Element deformation table  
MGG Partition of mass matrix; g set  
CASECC Case control data table

SLT, BGPDT, and SIL cannot be purged if external static loads or LOAD cards are selected in CASECC.

CSTM cannot be purged if any grid point or load references a coordinate system other than basic.

CASECC cannot be purged.

V. OUTPUT DATA BLOCKS:

PG Static load vector matrix giving static loads; g set

PG can never be purged.

VI. PARAMETERS:

LUSET Input; integer; no default. LUSET defines length of PG.

NSKIP Input; integer; no default. One static load is built for each CASECC record starting with NSKIP + 1, provided the boundary conditions are constant.

I. NAME: SSG2 (Static solution generator - phase 2)

II. PURPOSE: To reduce the applied load vectors and enforced displacements into equivalent load vectors applied to the independent displacement coordinate sets.

III. DMAP CALLING SEQUENCE:

SSG2 USET,YS,KFS,GØ,DM,PG/QR,PØ,PS,PL / \$

IV. INPUT DATA BLOCKS:

USET Displacement set definitions table

YS Constrained displacements; s set

KFS Partition of stiffness matrix after single-point constraints have been removed

GØ Structural matrix partitioning transformation matrix

DM Rigid-body transformation matrix

PG Static load vector matrix giving static loads; g set

USET must be present.

YS must be present if s set is not null.

KFS must be present if s set is not null.

GØ must be present if o set is not null.

DM must be present if r set is not null.

PG must be present.

V. OUTPUT DATA BLOCKS:

QR Determinate support forces matrix; r set

PØ Partition of the load vector matrix giving loads due to static force; o set

PS Partition of load vector matrix giving loads in s set

PL Partition of load vector matrix giving static loads on  
l set

QR must be present if r set is nonnull.

PØ must be present if o set is nonnull.

PS must be present if s set is nonnull.

PL must be present if l set is nonnull.

If the problem has no sets, SSG2 will return.

VI. PARAMETERS: None

I. NAME: SSG3 (Static solution generator - phase 3)

II. PURPOSE: To perform the actual static solutions. A displacement solution is produced for each applied load and tested for possible matrix decomposition errors.

III. DMAP CALLING SEQUENCE:

SSG3 LLL,  $\left\{ \begin{array}{l} \text{KLL} \\ \text{KAA} \end{array} \right\}$ , PL, L~~00~~, K~~00~~, P~~0~~/ULV, U~~00~~V, RULV, RU~~0~~V/V,N,~~0~~MIT/  
V,Y,IRES/V,N,NSKIP/V,N,EPSI \$

IV. INPUT DATA BLOCKS:

LLL Lower triangular factor of KLL; l set  
KLL Partition of stiffness matrix; l set  
KAA Partition of stiffness matrix; a set  
PL Partition of the load vector matrix giving static loads on l set  
L~~00~~ Lower triangular factor of K~~00~~; o set  
K~~00~~ Partition of stiffness matrix; o set  
P~~0~~ Partition of the load vector matrix giving loads due to static forces; o set

ULL, LLL, and PL must be present.

KLL can be purged if RULV is purged.

U~~00~~, L~~00~~, and P~~0~~ can be purged if ~~0~~MIT < 0.

K~~00~~ can be purged if ~~0~~MIT < 0 or RU~~0~~V is purged.

V. OUTPUT DATA BLOCKS:

ULV Partition of the displacement vector matrix giving displacements; l set  
U~~00~~V Partition of the displacement vector matrix giving displacements; o set

RULV            Residual vector matrix for the  $l$  set

RUØV           Residual vector matrix for the  $o$  set

ULV must be present.

UØØV can be purged if  $ØMIT < 0$ .

RULV and RUØV can be purged.

$$[RULV] = [KLL] [ULV] - [PL].$$

$$[RUØV] = [KØØ] [UØØV] - [PØ].$$

#### VI. PARAMETERS:

ØMIT            Input; integer; no default. ØMIT controls operations on  
o-set matrices.

IRES            Not used.

NSKIP           Input; integer; Default = 0. Identifies load vector numbers  
for diagnostic printout.

EPSI            Output; real; Default = 0.0. Value of total residual  
error  $\Sigma_e$  of last vector.

I. NAME: TAL (Table assembler)

II. PURPOSE: TAL processes element connection data, element property data, and geometry. These data are merged in two different sorts for efficiency in later processing. The element summary table contains connection, property, and geometry data for each element. The element connection and properties table contains, for each grid or scalar point in the model, connection, property, and geometry data for all elements connected to the point. Element temperature data are also included in the EST where applicable. The grid-point element connection table contains, for each grid point in the model, connection data for all elements connected to the point.

III. DMAP CALLING SEQUENCE:

```
TAL ECT,EPT,BGPDT,SIL,GPDT,CSTM / EST,,GEI,GPECT, / V,N,LUSET /  
V,N,NOSIMP / C,N,1 / V,N,NGENL / V,N,GENEL $
```

IV. INPUT DATA BLOCKS:

ECT	Element connection table
EPT	Element properties table
BGPDT	Basic grid-point definition table
SIL	Scalar index list
GPDT	Grid-point definition table
CSTM	Coordinate system transformation matrices

The ECT, BGPDT, and SIL data blocks may not be purged.

V. OUTPUT DATA BLOCKS:

EST	Element summary table
GEI	General element input
GPECT	Grid-point element connection table

VI. PARAMETERS:

LUSET        Input; integer; no default. Degrees of freedom in the  
              g displacement set

NØSIMP       Output; integer; no default. Number of elements in the  
              model (exclusive of general elements) or -1 if no elements.

NØGENL       Output; integer; -1

GENEL        Output; integer; no default. GENEL = -NØGENL.

## 5.4 Matrix Operation Modules

Module	Basic operation	Page
ADD	$[X] = a[A] + b[B]$	185
DECOMP	$[A] = > [L][U]$	186
FBS	$[X] = ([L][U])^{-1} [B]$	188
MERGE	$[A] < = \begin{bmatrix} A11 &   & A12 \\ \hline A21 &   & A22 \end{bmatrix}$	190
MPYAD	$[X] = [A][B] + [C]$	192
PARTN	$[A] = > \begin{bmatrix} A11 &   & A12 \\ \hline A21 &   & A22 \end{bmatrix}$	194
SOLVE	$[X] = [A]^{-1} [B]$	199
TRNSP	$[X] = [A]^T$	201



I. NAME: ADD (Matrix add)

II. PURPOSE: To compute  $[X] = a[A] + b[B]$  where a and b are scale factors.

III. DMAP CALLING SEQUENCE:

ADD A,B / X / C,Y,ALPHA=(1.0,2.0) / C,Y,BETA=(3.0,4.0) \$

IV. INPUT DATA BLOCKS:

A Any matrix

B Any matrix

[A] and/or [B] may be purged, in which case the corresponding term in the matrix sum will be assumed null. The input data blocks must be unique.

V. OUTPUT DATA BLOCKS:

X Matrix

The type of [X] is maximum of the types [A], [B], a, and b. The size of [X] is the size of [A] if [A] is present. Otherwise, it is that of [B].

[X] cannot be purged.

VI. PARAMETERS:

ALPHA Input; complex; single precision; Default = (1.0, 0.0).  
This is a, the scalar multiplier for [A].

BETA Input; complex; single precision; Default = (1.0, 0.0).  
This is b, the scalar multiplier for [B].

If  $\text{Im}(\text{ALPHA})$  or  $\text{Im}(\text{BETA}) = 0.0$ , the corresponding parameter will be considered real.

I. NAME: `DECOMP` (Matrix decomposition)

II. PURPOSE: To decompose a square matrix  $[A]$  into upper and lower triangular factors  $[U]$  and  $[L]$ .

$$[A] = [L][U]$$

III. DMAP CALLING SEQUENCE:

```
DECOMP A / L,U, / V,Y,KSYP / V,Y,CHOLSKY / V,N,MINDIAG / V,N,DET /  
V,N,POWER / V,N,SING $
```

IV. INPUT DATA BLOCKS:

A Square matrix

V. OUTPUT DATA BLOCKS:

L Nonstandard lower triangular factor of  $[A]$

U Nonstandard upper triangular factor of  $[A]$

VI. PARAMETERS:

KSYP Input; integer; Default = 1. If 1, use symmetric decomposition; if 0, use unsymmetric decomposition.

CHOLSKY Input; integer; Default = 0. If 1, use Cholesky decomposition - matrix must be positive definite; if 0, do not use Cholesky decomposition.

MINDIAG Output; real; double precision; Default = 0.0D0. The minimum diagonal term of  $[U]$ .

DET Output; complex; single precision; Default = 0.0D0. The scaled value of the determinant of  $[A]$ .

POWER Output; integer; Default = 0. Integer POWER of 10 by which DET should be multiplied to obtain the determinant of  $[A]$ .

SING Output; integer; Default = 0. SING is set to -1 if  $[A]$  is singular.

VII. REMARKS:

1. Nonstandard triangular factor matrix data blocks are used to improve the efficiency of the back substitution process in module FBS.
2. The matrix manipulating utility modules should be cautiously employed when dealing with nonstandard matrix data blocks.
3. If the CHOLSKY option is selected, the resulting factor (which will be written as  $[U]$ ) cannot be input to FBS.
4. Variable parameters output from functional modules must be saved if they are to be subsequently used. Replace V with S in parameter string to save parameter.

I. NAME: FBS (Matrix forward-backward substitution)

II. PURPOSE: To solve the matrix equation  $[L][U][X] = \pm[B]$  where  $[L]$  and  $[U]$  are the lower and upper triangular factors of a matrix previously obtained by means of functional module DECOMP.

III. DMAP CALLING SEQUENCE:

FBS L,U,B / X / V,Y,SYM / V,Y,SIGN / V,Y,PREC / V,Y,TYPE \$

IV. INPUT DATA BLOCKS:

L Nonstandard lower triangular factor

U Nonstandard upper triangular factor

B Rectangular matrix

V. OUTPUT DATA BLOCKS:

X Rectangular matrix having the same dimensions as  $[B]$

VI. PARAMETERS:

SYM Input; integer; Default = 0  $\left\{ \begin{array}{l} 1 \text{ matrix } [L][U] \text{ is symmetric} \\ -1 \text{ matrix } [L][U] \text{ is unsymmetric} \\ 0 \text{ reset to 1 or -1, depending} \\ \text{upon } [U] \text{ being purged or} \\ \text{not, respectively} \end{array} \right.$

Output; integer SYM used

SIGN Input; integer; Default = 1  $\left\{ \begin{array}{l} 1 \text{ solve } [L][U][X] = [B] \\ -1 \text{ solve } [L][U][X] = -[B] \end{array} \right.$

PREC Input; integer; Default = 0  $\left\{ \begin{array}{l} 1 \text{ use single-precision arithmetic} \\ 2 \text{ use double-precision arithmetic} \\ 0 \text{ logical choice based on input} \\ \text{and system precision flag} \end{array} \right.$

Output; integer Precision used

TYPE Input; integer; Default = 0

- 1 output type of matrix [X] is real single precision
- 2 output type of matrix [X] is real double precision
- 3 output type of matrix [X] is complex single precision
- 4 output type of matrix [X] is complex double precision
- 0 logical choice based on input matrices

Output; integer

TYPE used

VII. REMARKS:

1. Nonstandard triangular factor matrix data blocks are used to improve the efficiency of the back substitution process.
2. The matrix manipulating utility modules should be cautiously employed when dealing with nonstandard matrix data blocks.

I. NAME: MERGE (Matrix merge)

II. PURPOSE: To form the matrix [A] from its partitions:

$$[A] \leftarrow \begin{array}{c} \uparrow \\ \text{RP} \\ \downarrow \end{array} \left[ \begin{array}{cc|c} \leftarrow \text{CP} \rightarrow & & \\ \hline \text{A11} & \text{A12} & \\ \hline \text{A21} & \text{A22} & \\ \hline \end{array} \right] \begin{array}{l} = 0 \\ \neq 0 \end{array}$$

$= 0 \quad \neq 0$

III. DMAP CALLING SEQUENCE:

MERGE A11,A21,A12,A22,CP,RP / A / V,Y,SYM / V,Y,TYPE /  
V,Y,FORM \$

IV. INPUT DATA BLOCKS:

A11 Matrix

A21 Matrix

A12 Matrix

A22 Matrix

CP Column partitioning vector; single-precision column vector.  
(See the following comments.)

RP Row partitioning vector; single-precision column vector.  
(See the following comments.)

Any or all of [A11], [A12], [A21], and [A22] can be purged. When all are purged, it implies [A] = [0].

{RP} and {CP} may not both be purged.

See Remarks for meaning when either {RP} or {CP} is purged.

[A11], [A12], [A21], and [A22] must be unique matrices.

V. OUTPUT DATA BLOCKS:

A Merged matrix from [A11], [A12], [A21], and [A22]

[A] cannot be purged.

**VI. PARAMETERS:**

- SYM**            Input; integer; Default = -1. SYM < 0; {CP} is used for {RP}. SYM ≥ 0; {CP} and {RP} are distinct.
- TYPE**           Input; integer; Default = 0. Type of [A] (see remark (4)).
- FORM**           Input; integer; Default = 0. Form of [A] (see remark (3)).

**VII. REMARKS:**

1. MERGE is the inverse of PARTN in the sense that if [A11], [A12], [A21], and [A22] were produced by PARTN using {RP}, {CP}, FORM, SYM, and TYPE from [A], MERGE will produce [A]. See PARTN for options on {RP}, {CP}, and SYM.
2. All input data blocks must be distinct.
3. When FORM = 0, a compatible matrix [A] results as shown in the following table:

		Form of A22		
		Square	Rectangular	Symmetric
Form of All	Square	Square	Rectangular	Rectangular
	Rectangular	Rectangular	Rectangular	Rectangular
	Symmetric	Rectangular	Rectangular	Symmetric

4. If TYPE = 0, the type of the output matrix will be the maximum type of [A11], [A12], [A21], and [A22].

I. NAME: MPYAD (Matrix multiply and add)

II. PURPOSE: MPYAD performs the multiplication of two matrices and, optionally, addition of a third matrix to the product. By means of parameters, the user may compute  $\pm[A][B] \pm [C] = [X]$ , or  $\pm[A]^T[B] \pm [C] = [X]$ .

III. DMAP CALLING SEQUENCE:

MPYAD A,B,C / X / V,N,T / V,N,SIGNAB / V,N,SIGNC / V,N,PREC \$

IV. INPUT DATA BLOCKS:

A Left-hand matrix in the matrix product  $[A][B]$

B Right-hand matrix in the matrix product  $[A][B]$

C Matrix to be added to  $[A][B]$

If no matrix is to be added,  $[C]$  must be purged.

$[A]$ ,  $[B]$ , and  $[C]$  must be physically different data blocks.

$[A]$  and  $[B]$  must not be purged.

V. OUTPUT DATA BLOCKS:

X Matrix resulting from the MPYAD operation

$[X]$  may not be purged.

VI. PARAMETERS:

T Integer; input; no default  $T = \begin{cases} 1 & \text{perform } [A]^T[B] \\ 0 & \text{perform } [A][B] \end{cases}$

SIGNAB Integer; input; Default = 1  $SIGNAB = \begin{cases} +1 & \text{perform } [A][B] \\ 0 & \text{omit } [A][B] \\ -1 & \text{perform } -[A][B] \end{cases}$

SIGNC Integer; input; Default = 1  $SIGNC = \begin{cases} +1 & \text{add } [C] \\ 0 & \text{omit } [C] \\ -1 & \text{subtract } [C] \end{cases}$



PREC Integer; input; Default = 0

PREC =

- 1 elements of  $[X]$  will be output in single precision
- 2 elements of  $[X]$  will be output in double precision
- 0 logical choice based on input and system precision flag

### VII. EXAMPLES:

1.  $[X] = [A][B] + [C]$  ( $[X]$  see "Output Data Blocks")

MPYAD A,B,C / X / C,N,0 \$

2.  $[X] = [A]^T[B] - [C]$  ( $[X]$  single precision)

MPYAD A,B,C / X / C,N,1 / C,N,1 / C,N,-1 / C,N,1 \$

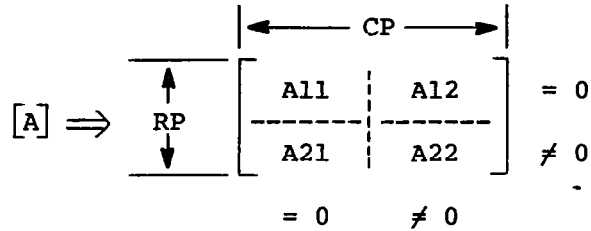
3.  $[X] = -[A][B]$  ( $[X]$  see "Output Data Blocks")

MPYAD A,B, / X / C,N,0 / C,N,-1 \$

The precision of  $[X]$  is determined from the input matrices in that, if any one of these matrices is specified as double precision, then  $[X]$  will also be double precision. If the precision for the input matrices is not specified, the precision of the system flag will be used.

I. NAME: PARTN (Matrix partition)

II. PURPOSE: To partition  $[A]$  into  $[A11]$ ,  $[A12]$ ,  $[A21]$ , and  $[A22]$ :



III. DMAP CALLING SEQUENCE:

PARTN A,CP,RP / A11,A21,A12,A22 / V,Y,SYM / V,Y,TYPE / V,Y,F11 /  
V,Y,F21 / V,Y,F12 / V,Y,F22 \$

IV. INPUT DATA BLOCKS:

A Matrix to be partitioned  
CP Column partitioning vector; single-precision column vector  
RP Row partitioning vector; single-precision column vector

V. OUTPUT DATA BLOCKS:

A11 Upper left partition of  $[A]$   
A21 Lower left partition of  $[A]$   
A12 Upper right partition of  $[A]$   
A22 Lower right partition of  $[A]$

Any or all output data blocks may be purged.

For size of outputs, see the METHOD section that follows.

VI. PARAMETERS:

SYM Input; integer; Default = -1. SYM chooses between a symmetric partition and one unsymmetric partition. If  $SYM < 0$ , {CP} is used as {RP}. If  $SYM \geq 0$ , {CP} and {RP} are distinct.  
TYPE Input; integer; Default = 0. Type of output matrices (see remark 8).

F11	Input; integer; Default = 0.	Form of [A11]	} See Remark 7
F21	Input; integer; Default = 0.	Form of [A21]	
F12	Input; integer; Default = 0.	Form of [A12]	
F22	Input; integer; Default = 0.	Form of [A22]	

VII. METHOD:

Let NC = number of nonzero terms in {CP}

Let NR = number of nonzero terms in {RP}

Let NRØWA = number of rows in [A]

Let NCØLA = number of columns in [A]

Case 1 {CP} purged and  $\text{SYM} \geq 0$

[A11] is an (NRØWA - NR) by NCØLA matrix

[A21] is an NR by NCØLA matrix

[A12] is not written

[A22] is not written

$$[A] \rightarrow \begin{bmatrix} A11 \\ \text{---} \\ A21 \end{bmatrix}$$

Case 2 {RP} purged and  $\text{SYM} \geq 0$

[A11] is an NRØWA by (NCØLA - NC) matrix

[A21] is not written

[A12] is an NRØWA by NC matrix

[A22] is not written

$$[A] \rightarrow \begin{bmatrix} A11 & | & A12 \end{bmatrix}$$

Case 3  $\text{SYM} < 0$  ({RP} must be purged)

[A11] is an (NRØWA - NC) by (NCØLA - NC) matrix

[A21] is an NC by (NCØLA - NC) matrix

[A12] is an (NRØWA - NC) by NC matrix

[A22] is an NC by NC matrix

$$[A] \rightarrow \begin{bmatrix} A11 & | & A12 \\ \text{---} & & \text{---} \\ A21 & | & A22 \end{bmatrix}$$

Case 4 Neither {CP} nor {RP} is purged and SYM > 0

[A11] is an (NRØWA - NR) by (NCØLA - NC) matrix

[A21] is an NR by (NCØLA - NC) matrix

[A12] is an (NRØWA - NR) by NC matrix .

[A22] is an NR by NC matrix

$$[A] \rightarrow \left[ \begin{array}{c|c} A11 & A12 \\ \hline A21 & A22 \end{array} \right]$$

#### VIII. REMARKS:

1. If [A] is purged, PARTN will cause all output data blocks to be purged.

2. If {CP} is purged, [A] is partitioned as follows:

$$[A] = > \left[ \begin{array}{c} A11 \\ \hline A21 \end{array} \right]$$

3. If {RP} is purged and SYM ≥ 0, [A] is partitioned as follows:

$$[A] = > \left[ \begin{array}{c|c} A11 & A12 \end{array} \right]$$

4. If {RP} is purged and SYM < 0, [A] is partitioned as follows:

$$[A] = > \left[ \begin{array}{c|c} A11 & A12 \\ \hline A21 & A22 \end{array} \right]$$

where {CP} is used as both the row and column partitioner.

5. {RP} and {CP} cannot both be purged.

6.

$$[A] = > \left[ \begin{array}{c|c} A11 & A12 \\ \hline A21 & A22 \end{array} \right]$$

Let [A] be an m by n order matrix.

Let {CP} be an n order row matrix containing q zero elements.

Let {RP} be an m order column vector containing p zero elements.

Partition of [A11] will consist of all elements  $A_{ij}$  of [A] for which  $CP_j = RP_i = 0$  in the same order as they appear in [A].

Partition  $[A12]$  will consist of all elements  $A_{ij}$  of  $[A]$  for which  $CP_j \neq 0$  and  $RP_i = 0$  in the same order as they appear in  $[A]$ .

Partition  $[A21]$  will consist of all elements  $A_{ij}$  of  $[A]$  for which  $CP_j = 0$  and  $RP_i \neq 0$  in the same order as they appear in  $[A]$ .

Partition  $[A22]$  will consist of all elements  $A_{ij}$  of  $[A]$  for which  $CP_j = 0$  and  $RP_i \neq 0$  in the same order as they appear in  $[A]$ .

7. If the defaults for F11, F21, F12, or F22 are used, the corresponding matrix will be output with a compatible form entered in the trailer.
8. If TYPE = 0, the type of the output matrices will be the type of the input matrix  $[A]$ .

#### IX. EXAMPLES:

1. Let  $[A]$ ,  $\{CP\}$ , and  $\{RP\}$  be defined as follows:

$$[A] = \begin{bmatrix} 1.0 & 2.0 & 3.0 & 4.0 \\ 5.0 & 6.0 & 7.0 & 8.0 \\ 9.0 & 10.0 & 11.0 & 12.0 \end{bmatrix} \quad \{CP\} = \begin{Bmatrix} 1.0 \\ 0.0 \\ 1.0 \\ 1.0 \end{Bmatrix} \quad \{RP\} = \begin{Bmatrix} 0.0 \\ 0.0 \\ 1.0 \end{Bmatrix}$$

Then, the DMAP instruction

```
PARTN A,CP,RP / A11,A21,A12,A22 / C,N,1 $
```

will create the real double-precision matrices

$$[A11] = \begin{bmatrix} 2.0 \\ 6.0 \end{bmatrix}, \quad F11 = 2 \quad [A12] = \begin{bmatrix} 1.0 & 3.0 & 4.0 \\ 5.0 & 7.0 & 8.0 \end{bmatrix}, \quad F12 = 2$$

$$[A21] = [10.0], \quad F21 = 1 \quad [A22] = [9.0 \ 11.0 \ 12.0], \quad F22 = 2$$

2. If, in example 1, the DMAP instruction were written as

```
PARTN A,CP, / A11,A21,A12,A22 / C,N,1 $
```

the resulting matrices would be

$$[A11] = \begin{bmatrix} 2.0 \\ 6.0 \\ 10.0 \end{bmatrix} \quad [A12] = \begin{bmatrix} 1.0 & 3.0 & 4.0 \\ 5.0 & 7.0 & 8.0 \\ 9.0 & 11.0 & 12.0 \end{bmatrix}$$

$$[A21] = \text{Purged} \quad [A22] = \text{Purged}$$

3. If, in example 1, the DMAP instructions were written as

```
PARTN A,,RP / A11,A21,A12,A22 / C,N,1 $
```

the resulting matrices would be

$$[A11] = \begin{bmatrix} 1.0 & 2.0 & 3.0 & 4.0 \\ 5.0 & 6.0 & 7.0 & 8.0 \end{bmatrix} \quad [A12] = \text{Purged}$$

$$[A21] = \begin{bmatrix} 9.0 & 10.0 & 11.0 & 12.0 \end{bmatrix} \quad [A22] = \text{Purged}$$

I. NAME: `SOLVE` (Linear system solver)

II. PURPOSE: To solve the matrix equation

$$[A] [X] = \pm [B]$$

III. DMAP CALLING SEQUENCE:

`SOLVE A,B / X / V,Y,SYM / V,Y,SIGN / V,Y,PREC / V,Y,TYPE $`

IV. INPUT DATA BLOCKS:

A Square, real, or complex matrix

B Rectangular, real, or complex matrix (if purged, the identity matrix is assumed)

V. OUTPUT DATA BLOCKS:

X A rectangular matrix

A standard matrix trailer will be written, identifying  $[X]$  as a rectangular matrix with the same dimensions as  $[B]$  and the type specified.

VI. PARAMETERS:

<code>SYM</code>	Input; integer; Default = 0	}	-1 use unsymmetric decomposition
			1 use symmetric decomposition
			0 logical choice based on input matrices
	Output; integer		SYM used

<code>SIGN</code>	Input; integer; Default = 1	}	1 solve $[A] [X] = [B]$
			-1 solve $[A] [X] = -[B]$

<code>PREC</code>	Input; integer; Default = 0	}	0 logical choice based on input
			1 use single-precision arithmetic
			2 use double-precision arithmetic
	Output; integer		PREC used

TYPE Input; integer; Default = 0

- 0 logical choice based on input
- 1 output type of matrix  $[X]$  is real single precision
- 2 output type of matrix  $[X]$  is real double precision
- 3 output type of matrix  $[X]$  is complex single precision
- 4 output type of matrix  $[X]$  is complex double precision

Output; integer

TYPE used

### VII. METHOD:

Depending on the SYM flag and the type of  $[A]$ , one of the subroutines SDCOMP or DECOMP is called to form  $[A] = [L][U]$ .

FBS is then called to solve  $[L][Y] = \pm[B]$  and  $[U][X] = [Y]$ , as appropriate.



I. NAME: TRNSP (Matrix transpose)

II. PURPOSE: To form  $[A]^T$  given  $[A]$ .

III. DMAP CALLING SEQUENCE:

TRNSP A/X \$

IV. INPUT DATA BLOCKS:

A Any matrix data block

If  $[A]$  is purged, TRNSP will cause  $[X]$  to be purged.

V. OUTPUT DATA BLOCKS:

X Matrix transpose of  $[A]$

$[X]$  cannot be purged.

VI. PARAMETERS: None

VII. REMARKS:

1. Transposition of large full matrices is very expensive and should be avoided if possible.
2. TRNSP currently uses an algorithm that assumes the matrix is dense. This algorithm is extremely inefficient for sparse matrices. Sparse matrices should be transposed by using MPYAD.

## 5.5 Utility Modules

Module	Basic function	Page
COPY	Define a new data block as a physical copy of one already existing	203
DIAGONAL	Strip diagonal from matrix	204
INPUTT2	Read data blocks from FORTRAN-written user tapes	205
MATPRN	Print matrices	207
OUTPUT2	Write data blocks onto user tapes by means of FORTRAN	208
PARAM	Manipulate parameter values	210
PARAML	Select parameters from a user input matrix or table	212
PARAMR	Perform specified arithmetic, logical, and conversion operations on real or complex parameters	213
PRTPARAM	Print parameter values and DMAP error	215
SCALAR	Convert matrix element to parameter	216
SETVAL	Set parameter values	217
SWITCH	Interchange the data block names of two files	218

Utility modules are an arbitrary subdivision of the functional modules and are used to output matrix and table data blocks and to manipulate parameters.

I. NAME: COPY (Define new physical copy of existing data block)

II. PURPOSE: To generate a physical copy of a data block.

III. DMAP CALLING SEQUENCE:

COPY DB1 / DB2 / PARAM \$

IV. INPUT DATA BLOCKS:

DB1 Any NASTRAN data block

V. OUTPUT DATA BLOCKS:

DB2 Any valid NASTRAN data block name

VI. PARAMETERS:

PARAM If  $PARAM < 0$ , the copy will be performed - integer; input;  
no default

VII. METHOD: If  $PARAM \geq 0$ , a return is made; otherwise, a physical copy of the input data block is generated.

VIII. REMARKS:

1. The input data block may not be purged.

- I. NAME: `DIAGONAL` (Strip diagonal from matrix)
- II. PURPOSE: To remove the real part of the diagonal from a matrix, raise each term to a specified power, and output a column vector or square symmetric matrix.
- III. DMAP CALLING SEQUENCE:
- `DIAGONAL A/B/C,Y,OPT=COLUMN/V,Y,POWER=1. $`
- IV. INPUT DATA BLOCKS:
- A Any square or diagonal matrix
- V. OUTPUT DATA BLOCKS:
- B Either a real column vector or symmetric matrix containing the diagonal of A
- VI. PARAMETERS:
- `OPT` Input; BCD; Default = `COLUMN`
- = '`COLUMN`' produces column vector output (labeled as a general rectangular matrix)
- = '`SQUARE`' produces square matrix (labeled a symmetric matrix)
- `POWER` Input; real, single precision; Default = 1. Exponent to which the real part of each diagonal element is raised.
- VII. REMARKS:
1. The module checks for special cases of `POWER = 0., 0.5, 1.0, and 2.`
  2. The precision of the output matrix matches the precision of the input matrix.

I. NAME: INPUTT2 (Reads user-written FORTRAN tapes)

II. PURPOSE: Recovers up to five data blocks from a FORTRAN-written user tape. This tape may be written either by a user-written FORTRAN program or by the companion module OUTPUT2.

III. DMAP CALLING SEQUENCE:

INPUTT2 / DB1,DB2,DB3,DB4,DB5 / V,N,P1 / V,N,P2 / V,N,P3 \$

IV. INPUT DATA BLOCKS:

Input data blocks are not used in this module call statement.

V. OUTPUT DATA BLOCKS:

DBi Data blocks which will be recovered from one of the NASTRAN FORTRAN tape files UT1, UT2 to UT5. Any or all of the output data blocks may be purged. Only nonpurged data blocks will be taken from the tape. The data blocks will be taken sequentially from the tape, starting from a position determined by the value of the first parameter. Note that the output data block sequence A,B,,, is the same as ,A,,B, or ,,,A,B .

VI. PARAMETERS: The meaning of the first parameter (P1) value is given in the table below. (The default value is 0.)

P1 value	Meaning
+n	Skip forward n data blocks before reading.
0	Data blocks are read starting at the current position. The current position for the first use of a tape is at the label P3. Hence, P3 counts as one data block.
-1	Rewind before reading; position tape past label P3.
-3	Print data block names and then rewind before reading.
-5	Search user tape for first version of data block (DBi) requested. If any (DBi) are not found, fatal termination occurs.
-6	Search user tape for final version of data block (DBi) requested. If any (DBi) are not found, fatal termination occurs.
-7	Search user tape for first version of data block (DBi) requested. If any (DBi) are not found, warning message is written on the output file and run continues.
-8	Search user tape for final version of data block (DBi) requested. If any (DBi) are not found, warning message is written on the output file and run continues.

The second parameter P2 for this module is the FORTRAN unit number from which the data blocks will be read. This unit is not required to be a physical tape. The allowable values for this parameter are highly machine and installation dependent. (The default value for P2 is 0.)

User tape code	FORTRAN file name
11	UT1
12	UT2
13	UT3
14	UT4
15	UT5

The third parameter P3 for this module is used as the FORTRAN user tape label for NASTRAN identification. The label P3 is an alphanumeric variable of eight characters or less (the first character must be alphabetic). The value of P3 must match a corresponding value on the NASTRAN user tape. The comparison of P3 with the value on the user tape is dependent on the value of P1 as shown in the table below. (The default value for P3 is XXXXXXXX.)

P1 value	Tape label checked
+n	No
0	No
-1	Yes
-3	Yes (warning check)
-5	Yes
-6	Yes
-7	Yes
-8	Yes

I. NAME: MATPRN (General matrix printer)

II. PURPOSE: To print general matrix data blocks.

III. DMAP CALLING SEQUENCE:

MATPRN M1,M2,M3,M4,M5 // \$

IV. INPUT DATA BLOCKS:

Mi Matrix data blocks, any of which may be purged

V. OUTPUT DATA BLOCKS: None

VI. PARAMETERS: None

VII. OUTPUT:

The nonzero band of each column of each input matrix data block is unpacked and printed in single precision.

VIII. REMARKS:

1. Any or all input data blocks can be purged.

IX. EXAMPLES:

1. MATPRN KGG,,,, // \$

2. MATPRN KGG,PL,PG,BGG,UPV // \$

I. NAME: `OUTPUT2` (Creates user-written FORTRAN tapes)

II. PURPOSE: Writes up to five data blocks and a user tape label onto a FORTRAN-written user tape for subsequent use at a later date with the companion module INPUTT2. OUTPUT2 is also used to position the user tape prior to writing the data blocks. Multiple calls are allowed. A message is written on the output file for each data block successfully written. The user is cautioned to be careful when positioning a user tape with OUTPUT2 since he may inadvertently destroy information through improper positioning. Even though no data blocks are written, an EOF, which has the effect of destroying anything on the tape forward of the current position, will be written at the completion of each call.

III. DMAP CALLING SEQUENCE:

`OUTPUT2 DB1,DB2,DB3,DB4,DB5 // V,N,P1 / V,N,P2 / V,N,P3 $`

IV. INPUT DATA BLOCKS:

DBi Any data block which the user desires to be written on one of the NASTRAN FORTRAN tape files UT1, UT2 to UT5. Any or all of the input data blocks may be purged. Only nonpurged data blocks will be placed on the tape.

V. OUTPUT DATA BLOCKS: None

VI. PARAMETERS:

The meaning of the first parameter P1 value is given in the following table. (The default value is 0.)

P1 value	Meaning
+n	Skip forward n data blocks before writing.
0	Data blocks are written starting at the current position. The current position for the first use of a tape is at the label P3. In this case, P3 counts as one data block.
-1	Rewind before writing.
-3	Rewind tape; print data block names; and then write after the last data block on the tape.
-9	Write a final EOF on the tape.



The second parameter P2 for this module is the FORTRAN unit number, onto which the data blocks will be written. This unit is not required to be a physical tape. The allowable values for this parameter are highly machine or installation dependent. (The default value for P2 is 0.)

User tape code	FORTRAN file name
11	UT1
12	UT2
13	UT3
14	UT4
15	UT5

The third parameter P3 for this module is used to define the FORTRAN user tape label. The label is used for NASTRAN identification. The label P3 is an alphanumeric variable of eight or less characters (the first character must be alphabetic) which is written on the user tape. The writing of this label is dependent on the value of P1 as follows (the default value for P3 is XXXXXXXX):

P1 value	Table label written
+n	No
0	No
-1	Yes
-3	No (warning check)
-9	No

The user may specify the third parameter as V,Y,name. The user then must also include a PARAM card in the bulk data deck to set a value for name.

## VII. REMARKS:

1. The primary objective of this module is to write tapes using simple FORTRAN so that a user can read NASTRAN generated data with his own program. Similarly, matrices can be generated with externally written simple FORTRAN programs and then read by module INPUTT2. To do this, the format of the information on these tapes must be adhered to. The basic idea is that a one-word logical KEY record is written which indicates what follows. A zero value indicates an end-of-file condition. A negative value indicates the end of a record where the absolute value is the record number. A positive value indicates that the next record consists of that many words of data.

- I. NAME: PARAM (Parameter processor)
- II. PURPOSE: To perform specified operations on integer DMAP parameters.
- III. DMAP CALLING SEQUENCE:
- PARAM // C,N,ØP / V,N,ØUT / V,N,IN1 / V,N,IN2 \$
- IV. INPUT DATA BLOCKS: None
- V. OUTPUT DATA BLOCKS: None
- VI. PARAMETERS:

- ØP            BCD operation code from the table below (input; no default);  
ØP is usually specified as a "C,N" parameter
- ØUT           Parameter which is being generated by PARAM (output;  
integer; Default = 1)
- IN1           Name of a parameter whose value is used to compute ØUT  
according to the table that follows (input; integer;  
Default = 1)
- IN2           Name of a parameter whose value is used to compute ØUT  
according to the table that follows (input; integer;  
Default = 1)

VII. REMARKS:

1. The following table gives the results for ØUT as a function of ØP, IN1, and IN2:

Arithmetic operations					
ØP	ADD	SUB	MPY	DIV	NOT
ØUT	IN1+IN2	IN1-IN2	IN1·IN2	IN1/IN2	-IN1

Logical operations

ØP	AND				ØR				IMPL			
ØUT	-1	+1	+1	+1	-1	-1	-1	+1	-1	+1	-1	-1
IN1	<0	<0	≥0	≥0	<0	<0	≥0	≥0	<0	<0	≥0	≥0
IN2	<0	≥0	<0	≥0	<0	≥0	<0	≥0	<0	≥0	<0	≥0

Special operations	
ØP	ØUT
NØP	ØUT (unchanged)
KLØCK	Current CPU time in integer seconds from the start of the job
TMTØGØ	Remaining CPU time in integer seconds based on the TIME card
PREC	Returns the currently requested precision; 2 = D.P. , 1 = S.P.

PARAM does its own save; therefore, an s need not replace the V in the calling sequence.

#### VIII. EXAMPLES:

PARAM // C,N,NØT / V,N,XYZ / V,N,NØXYZ \$ - this example changes the sense of parameter NØXYZ which may be useful for the CØND or EQUIV instructions. Alternatively, XYZ could have been set in the following ways:

PARAM // C,N,MPY / V,N,XYZ / V,N,NØXYZ / C,N,-1 \$

PARAM // C,N,IMPL / V,N,ABC / V,N,DEF / V,N,GHI \$

PARAM // C,N,NØP / V,N,P1=5 \$ - this example sets the value of parameters P1 to P5 and saves it for subsequent use.

I. NAME: PARAML (Selects parameters from a list)

II. PURPOSE: To select parameters from a user input matrix or table.

III. DMAP CALLING SEQUENCE:

```
PARAML INPUT // C,N,ØP / V,N,RECNØ / V,N,WØRDN /  
V,N,REAL1 / V,N,INTEG / V,N,REAL2 / V,N,BCD $
```

IV. INPUT DATA BLOCKS:

INPUT Any matrix or table

V. OUTPUT DATA BLOCKS: None

VI. PARAMETERS:

ØP	Input; BCD; no default
RECNØ	Input; integer; Default = 1
WØRDN	Input; integer; Default = 1
REAL1	Output; real; Default = 1.0
INTEG	Output; integer; Default = 0
REAL2	Output; real; Default = 1.0
BCD	Output; BCD; Default = Blank

VII. REMARKS:

1. REAL1, INTEG, REAL2, and BCD will be set by the module whenever they are V type parameters.
2. RECNØ and WØRDN control the starting point, according to ØP. If ØP = DMI, RECNØ is the column number and WØRDN is the row number. If ØP = DTI, RECNØ is the record number and WØRDN is the word number. If ØP = PRESENCE, INTEG will be -1 if INPUT is purged.
3. PARAML does its own save; therefore, an S need not replace a V in the calling sequence.

VIII. EXAMPLE:

Obtain the value in column 1, row 1 of a matrix.

```
PARAML KGG // C,N,DMI / C,N,1 / C,N,1 / V,N,TERM $
```

I. NAME: PARAMR (Parameter processor - real)

II. PURPOSE: To perform specified arithmetic, logical, and conversion operations on real or complex parameters.

III. DMAP CALLING SEQUENCE:

```
PARAMR // C,N,ØP / V,N,ØUTR / V,N,INR1 / V,N,INR2 /  
        V,N,ØUTC / V,N,INCL / V,N,INC2 /  
        V,N,FLAG $
```

IV. INPUT DATA BLOCKS: None

V. OUTPUT DATA BLOCKS: None

VI. PARAMETERS:

ØP	Input; BCD operation code from the table that follows; no default
ØUTR	Output; real; Default = 0.0
INR1	Input; real; Default = 0.0
INR2	Input; real; Default = 0.0
ØUTC	Output; complex; Default = (0.0,0.0)
INCL	Input; complex; Default = (0.0,0.0)
INC2	Input; complex; Default = (0.0,0.0)
FLAG	Output; integer; Default = 0

The values of the parameters are dependent upon ØP, as shown in the following table:

$\emptyset P$	$\emptyset UTPUTS$	$\emptyset P$	$\emptyset UTPUTS$
ADD	$\emptyset UTR = INR1 + INR2$	DIVC	$\emptyset UTC = INC1 / INC2$
SUB	$\emptyset UTR = INR1 - INR2$	CSQRT	$\emptyset UTC = \sqrt{INC1}$
MPY	$\emptyset UTR = INR1 * INR2$	C $\emptyset$ MPLEX	$\emptyset UTC = (INR1, INR2)$
DIV	$\emptyset UTR = INR1 / INR2$	C $\emptyset$ NJ	$\emptyset UTC = \overline{INC1}$
N $\emptyset$ P	RETURN	REAL	INR1 = Re ( $\emptyset UTC$ ) INR2 = Im ( $\emptyset UTC$ )
SQRT	$\emptyset UTR = \sqrt{INR1}$	EQ	FLAG = -1 if INR1 = INR2
SIN	$\emptyset UTR = SIN(INR1)$	GT	FLAG = -1 if INR1 > INR2
C $\emptyset$ S	$\emptyset UTR = C\emptyset S(INR1)$	LT	FLAG = -1 if INR1 < INR2
ABS	$\emptyset UTR =   INR1  $	LE	FLAG = -1 if INR1 $\leq$ INR2
EXP	$\emptyset UTR = \exp(INR1)$	GE	FLAG = -1 if INR1 $\geq$ INR2
TAN	$\emptyset UTR = TAN(INR1)$	NE	FLAG = -1 if INR1 $\neq$ INR2
N $\emptyset$ RM	$\emptyset UTR =    \emptyset UTC   $	L $\emptyset$ G	$\emptyset UTR = L\emptyset G_{10}(INR1)$
P $\emptyset$ WER	$\emptyset UTR = INR1 ** INR2$	LN	$\emptyset UTR = L\emptyset G_e(INR1)$
ADDC	$\emptyset UTC = INC1 + INC2$	FIX	FLAG = FIX ( $\emptyset UTR$ )
SUBC	$\emptyset UTC = INC1 - INC2$	FL $\emptyset$ AT	$\emptyset UTR = FL\emptyset AT(FLAG)$
MPYC	$\emptyset UTC = INC1 * INC2$		

### VII. REMARKS:

1. Any output parameter must be V type if the parameter is used by  $\emptyset P$  as output.
2. For  $\emptyset P = DIV$  or  $\emptyset P = DIVC$ , the output is zero if the denominator is zero.
3. PARAMR does its own save; therefore, an S need not replace a V in the calling sequence.
4. For  $\emptyset P = SIN$ ,  $\emptyset P = C\emptyset S$ , or  $\emptyset P = TAN$ , the input must be expressed in radians.

I. NAME: PRTPARM (Parameter and DMAP message printer)

II. PURPOSE: Prints parameter values and DMAP messages.

III. DMAP CALLING SEQUENCE:

PRTPARM // C,N,a / C,N,b / C,N,c \$

IV. INPUT DATA BLOCKS: None

V. OUTPUT DATA BLOCKS: None

VI. PARAMETERS:

a Integer value (no default value)

b BCD value (Default value = XXXXXXXX)

c Integer value (Default value = 0)

VII. METHOD:

As a parameter printer, use  $a = 0$ . There are two options:

b parameter name will cause the printout of the value of that parameter (Example: PRTPARM // C,N, $\emptyset$  / C,N,LUSET \$)

or

b XXXXXXXX will cause the printout of the values of all parameters in the current variable parameter table. Since this is the default value, it need not be specified (Example: PRTPARM / C,N, $\emptyset$  \$)

As a DMAP message printer, use  $a \neq 0$ . There are two options:

$a > 0$  causes the printout of the  $j$ th message of category  $b$  where  $j = |a|$  and  $b$  is one of the values shown below. (The number of messages available in each category is also given.) (Example: PRTPARM // C,N,1 / C,N,DMAP \$)

or

$a < 0$  causes the same action as  $a > 0$  with the additional action of program termination. Thus, PRTPARM may be used as a fatal message printer. (Example: PRTPARM // C,N,-2 / C,N,PLA \$)

VIII. REMARKS:

1.  $b$  is always a value.

2. Meaningless values of  $a$  and  $b$  result in diagnostic messages from PRTPARM.

I. NAME: SCALAR (Convert matrix element to parameter)

II. PURPOSE: To extract a specified element from a matrix for use as a parameter.

III. DMAP CALLING SEQUENCE:

SCALAR A//V,Y,NROW=1/V,N,NCOL=1/C,Y,VALUE \$

IV. INPUT DATA BLOCKS:

A May be any type of matrix

If A is purged, value will be returned as (0.,0.).

V. OUTPUT DATA BLOCKS: None

VI. PARAMETERS:

NROW Input; integer; Default = 1. Row number of element to be extracted from [A].

NCOL Input; integer; Default = 1. Column identification of element.

VALUE Output; complex; single precision; Default = (0.,0.). Contents of element (NROW,NCOL) in matrix [A].



I. NAME: SETVAL (Set values)

II. PURPOSE: Set DMAP parameter variable values equal to other DMAP parameter variables or DMAP parameter constants.

III. DMAP CALLING SEQUENCE:

```
SETVAL // V,N,X1 / V,N,A1 /  
        V,N,X2 / V,N,A2 /  
        V,N,X3 / V,N,A3 /  
        V,N,X4 / V,N,A4 /  
        V,N,X5 / V,N,A5 $
```

IV. INPUT DATA BLOCKS: None

V. OUTPUT DATA BLOCKS: None

VI. PARAMETERS:

X1, X2, X3, X4, X5      Output; integers; variables

A1, A2, A3, A4, A5      Input; integers; Default values = 1; variables  
or constants

VII. METHOD: This module sets X1 = A1, X2 = A2, X3 = A3, X4 = A4, and X5 = A5. Only two parameters need be specified in the calling sequence (X1 and A1).

VIII. REMARKS:

1. An S must replace the V in the SETVAL instruction if the output parameter values are to be subsequently used.
2. See PARAM for an alternate method of defining parameter values.
3. As an example, the statements

```
SETVAL // V,N,X1 / S,N,A1 / S,N,X2 / C,N,3 $
```

are equivalent to the statements

```
PARAM // C,N,ADD / V,N,X1 / V,N,A1 / C,N,0 $
```

```
PARAM // C,N,NOP / V,N,X2=3 $
```

I. NAME: SWITCH (Interchange two data block names)

II. PURPOSE: To interchange two data block names.

III. DMAP CALLING SEQUENCE:

SWITCH DB1,DB2 // PARAM \$

IV. INPUT DATA BLOCKS:

DB1 } Any NASTRAN data blocks  
DB2 }

V. OUTPUT DATA BLOCKS: None

VI. PARAMETERS:

PARAM If  $PARAM < 0$ , the switch is performed.

VII. METHOD: If  $PARAM \geq 0$ , a return is made; otherwise, the names of the data blocks are interchanged. All attributes of the data within the blocks remain constant; only the names are changed.

VIII. REMARKS:

1. Neither input data block may be purged.

2. This option is of use in iterative DMAP operations.

## 6. ERROR MESSAGES

### 6.1 NASTRAN System and User Messages

NASTRAN system and user messages are identified by number. Message numbers have been assigned in groups as follows:

1 - 1000	Preface messages
1001 - 2000	Executive module messages
2001 -	Functional module messages

These messages have the following format:

\*\*\* { SYSTEM } { POTENTIALLY FATAL } MESSAGE id, text.  
          { USER } { FATAL }  
                  { WARNING }  
                  { INFORMATION }

where id is a unique message identification number and text is the message, as indicated in capital letters, for each of the diagnostic messages. A series of asterisks (\*\*\*\*) in the text indicates information that will be filled in for a specific use of the message, such as the number of a grid point or the name of a bulk data card. Many of the messages are followed by additional explanatory material, including suggestions for remedial action.

The system and user messages described in this section pertain only to those messages generated by NASTRAN that are not self-explanatory, that have not been cut out when reducing NASTRAN's size, and that are useful to the user. Although these messages can appear at various places in the output stream, they should be easily identified by their format. The various computer operating systems also produce diagnostic messages that can appear at various places in the output stream. The format of these messages varies with the operating system. Reference should be made to the operating system manuals for interpreting the messages that are not generated by NASTRAN.

System messages refer to diagnostics that are associated with program errors. In general, such errors cannot be corrected by the user. User messages refer to errors that are usually associated with the preparation of the NASTRAN data deck. Corrective action is indicated in either the message text or the explanatory information following the text.

Fatal messages cause termination of the execution following printing of the message text. These messages always appear at the end of the NASTRAN output. Warning and information messages appear at various places in the output stream. Such messages only convey warnings or information to the user. Consequently, the execution continues in a normal manner following the printing of the message text.

As an example, consider message number 2025, which appears in the printed output as follows:

```
*** USER FATAL MESSAGE 2025, UNDEFINED COORDINATE SYSTEM 102.
```

The three leading asterisks (\*\*\*) are always present in user and system diagnostic messages. The word USER indicates that this is a user message rather than a system message. The word FATAL indicates that this is a fatal message rather than a warning or information message. The number 2025 is the identification number for this message. The text of the message follows the comma. The number 102 replaces the asterisks (\*\*\*\*) in the general message text and indicates that 102 is the identification number of the undefined coordinate system.

## 6.2 Preface User Messages

Preface messages indicate that an error has been found before or during the input of data.

```
01 *** USER WARNING MESSAGE 1, ASSUMED FIRST INPUT FILE IS NULL.
```

User has specified N input data blocks when there should be N+1.

```
02 *** USER WARNING MESSAGE 2, PARAMETER NAMED ***** IS DUPLICATED.
```

No harm done. Parameter is saved just once.

```
03 *** USER FATAL MESSAGE 3, FØR MAT ERRØR IN PARAMETER NØ.***.
```

Double delimiter appears in parameter section of previous DMAP instruction.

```
05 *** USER FATAL MESSAGE 5, PARAMETER INPUT DATA ERRØR ILLEGAL VALUE FØR  
PARAMETER NAMED *****.
```

Type of parameter on PARAM card is inconsistent with type of parameter by same name in previous DMAP instruction.

```
07 *** USER FATAL MESSAGE 7, PARAMETER NØ.*** NEEDS PARAMETER NAME.
```

Parameter is not in correct format.

```
08 *** USER FATAL MESSAGE 8, BULK DATA PARAM CARD ERRØR. MUST NØT DEFINE  
PARAMETER NAMED *****.
```

The "N" in V,N,\*\*\*\*\* means user cannot set the value of the parameter with name \*\*\*\*\* on a PARAM card.

```
09 *** USER FATAL MESSAGE 9, VALUE NEEDED FØR PARAMETER NØ. ***.
```

Constant needs value in DMAP instruction or on PARAM card.

12 \*\*\* USER FATAL MESSAGE 12, ILLEGAL CHARACTER IN DMAP INSTRUCTION NAME.

Name must be eight or less alphanumeric characters, the first character being alphabetic.

15 \*\*\* USER FATAL MESSAGE 15, INCONSISTENT LENGTH USED FOR PARAMETER NAMED \*\*\*\*\*.

This parameter was used in a previous DMAP instruction that gave it a different type.

18 \*\*\* USER FATAL MESSAGE 18, TOO MANY PARAMETERS IN DMAP PARAMETER LIST.

Incorrect calling sequence for DMAP instruction.

19 \*\*\* USER FATAL MESSAGE 19, LABEL NAMED \*\*\*\*\* IS MULTIPLY DEFINED.

LABEL named appears in more than one place in DMAP program.

20 \*\*\* USER FATAL MESSAGE 20, ILLEGAL CHARACTERS IN PARAMETER NO. \*\*\*.

Name must be eight or less alphanumeric characters, the first character being alphabetic.

23 \*\*\* USER FATAL MESSAGE 23, DATA BLOCK NAMED \*\*\*\*\* IS NOT REFERENCED IN SUBSEQUENT FUNCTIONAL MODULE.

Error can be suppressed by adding the following:

```
PARAM //C,N,NOP/V,N,TRUE=-1 $
```

```
COND LABELXXX,TRUE $
```

```
TABPT *****,,// $
```

```
LABEL LABELXXX $
```

25 \*\*\* USER FATAL MESSAGE 25, PARAMETER NAMED \*\*\*\*\* NOT DEFINED.

Parameter is referenced in nonfunctional module but is nowhere defined.

26 \*\*\* USER FATAL MESSAGE 26, LABEL NAMED \*\*\*\*\* NOT DEFINED.

LABEL name does not appear in LABEL instruction.

27 \*\*\* USER WARNING MESSAGE 27, LABEL NAMED \*\*\*\*\* NOT REFERENCED.

LABEL name appears only in a LABEL instruction.

37 \*\*\* USER WARNING MESSAGE 37, WARNING ONLY - MAY NOT BE ENOUGH FILES AVAILABLE FOR MODULE REQUIREMENTS. FILES NEEDED = \*\*\* FILES AVAILABLE = \*\*\*.

Program will execute if enough data blocks referenced by the module are purged. Purged data blocks are not assigned files.

38 \*\*\* SYSTEM FATAL MESSAGE 38, NOT ENOUGH CORE FOR GPI TABLES.

User must break up DMAP program.

42 \*\*\* USER WARNING MESSAGE 42, PARAMETER NAMED \*\*\*\*\* ALREADY HAD VALUE ASSIGNED PREVIOUSLY.

Parameter appears in a previous instruction which assigned it a value. The previous value will be used.

51 \*\*\* SYSTEM FATAL MESSAGE 51, NOT ENOUGH OPEN CORE FOR XGPIBS ROUTINE.

Additional core memory is required.

204 \*\*\* USER FATAL MESSAGE 204, COLD START, NO BULK DATA.

No data cards were found after the BEGIN BULK card. A blank card will satisfy this rule.

206 \*\*\* USER FATAL MESSAGE 206, PREVIOUS CONTINUATION MNEMONIC HAS A DUPLICATE.

Two or more continuation cards were found with columns 2 to 8 identical.

207 \*\*\* USER INFO MESSAGE 207, BULK DATA NOT SORTED, XSORT WILL REORDER DECK.

Bulk data deck was not in alphanumeric sort. Sorting will be performed. Sorting of large deck can be time consuming.

208 \*\*\* USER FATAL MESSAGE 208, PREVIOUS CARD IS A DUPLICATE PARENT.

Two or more cards were found with columns 74 to 80 identical, and a continuation card is present with that mnemonic (columns 2 to 8).

209 \*\*\* USER FATAL MESSAGE 209, PREVIOUS \*\*\*\* CONTINUATION CARDS HAVE NO PARENTS.

One or more continuation cards were found with a mnemonic (columns 2 to 8) not matching any other card (columns 74 to 80).

210 \*\*\* SYSTEM FATAL MESSAGE 210, SCRATCH COULD NOT BE OPENED.

One of the required scratch files was not present (destroyed) in FIST.

211 \*\*\* SYSTEM FATAL MESSAGE 211, ILLEGAL EØR ØN SCRATCH.

A required scratch file was formatted improperly.

216 \*\*\* SYSTEM FATAL MESSAGE 216, ILLEGAL INDEX.

FØRTRAN computed-GØ-TØ have received an illogical value.

300 \*\*\* USER FATAL MESSAGE 300, DATA ERRØR IN FIELD UNDERLINED.

(1)

A data error as described in the text has been detected by utility routine XRCARD or RCARD.

300 \*\*\* USER FATAL MESSAGE 300, INVALID DATA CØLUMN 72.

(2)

Error in format of exponent.

300 \*\*\* USER FATAL MESSAGE 300, INTEGER DATA ØUT ØF MACHINE RANGE.

(3)

The limits are  $2^{31} - 1$  for IBM,  $2^{59} - 1$  for Control Data, and  $2^{35} - 1$  for UNIVAC.

300 \*\*\* USER FATAL MESSAGE 300, INVALID CHARACTER FØLLØWING INTEGER IN

(4)

CØLUMN \*\*\*.

Either an illegal delimiter was detected or a real number is missing the decimal.

300 \*\*\* USER FATAL MESSAGE 300, DATA ERRØR - UNANTICIPATED CHARACTER IN

(5)

CØLUMN \*\*\*.

A  $\pm E$  or  $\pm D$  was expected, based on other input data.

300 \*\*\* USER FATAL MESSAGE 300, DATA ERRØR - MISSING DELIMITER ØR REAL PØWER

(6)

ØUT ØF MACHINE RANGE.

Either no delimiter was found or the power was exceeded. The limits are E-78 to E+75 for IBM, E-38 to E+38 for UNIVAC, and E-294 to E+322 for Control Data.

305 \*\*\* SYSTEM FATAL MESSAGE 305, GINØ CANNØT ØPEN FILE \*\*\*\*\*.

Unexpected nonstandard return from ØPEN.

306 \*\*\* SYSTEM FATAL MESSAGE 306, READ LØGIC RECØRD ERRØR.

Short record encountered. Bulk data card images occupy 20 words.

311 \*\*\* USER FATAL MESSAGE 311, NONUNIQUE FIELD 2 ON BULK DATA  
CARD \*\*\*\*\* \*\*\*.

The sorted bulk data card indicated must have a unique integer in  
field 2.

314 \*\*\* SYSTEM FATAL MESSAGE 314, INVALID CALL FROM IFP \*\*\*\*\*.

Code error, machine failure, or cell is being destroyed.

321 \*\*\* USER FATAL MESSAGE 321, NONUNIQUE PARAM NAME \*\*\*\*\*.

All names of parameters must be unique.

324 \*\*\* USER WARNING MESSAGE 324, BLANK CARD(S) IGNORED.

Blank bulk data cards are ignored by NASTRAN.

330 \*\*\* SYSTEM FATAL MESSAGE 330, NO ROOM IN CORE FOR PARAM CARDS.

Increase core size.

505 \*\*\* USER FATAL MESSAGE 505, CONTROL CARD \*\*\*\* IS ILLEGAL.

Card preceding message 505 cannot be processed correctly.

506 \*\*\* USER FATAL MESSAGE 506, CONTROL CARD \*\*\*\* DUPLICATED.

Card preceding message 506 cannot be input more than once.

511 \*\*\* SYSTEM FATAL MESSAGE 511, DMAP SEQUENCE EXCEEDS CORE SIZE -  
REMAINING DMAP INSTRUCTIONS IGNORED.

You have run out of open core. Split the DMAP sequence somewhere prior  
to where message 511 was printed out.

515 \*\*\* USER FATAL MESSAGE 515, END INSTRUCTION MISSING IN DMAP SEQUENCE.

DMAP sequence must end with END control card.

520 \*\*\* USER FATAL MESSAGE 520, CONTROL CARD \*\*\*\* IS MISSING.

The control card mentioned is required for this problem.

524 \*\*\* SYSTEM FATAL MESSAGE 524, ALTERNATE RETURN TAKEN WHEN OPENING FILE \*\*\*\*\*.

This occurs if file name is not in FIST or if the end of tape was  
reached while writing on the file. The file name should correspond  
to one of the permanent entries in the FIST.



525 \*\*\* SYSTEM FATAL MESSAGE 525, ILLEGAL FORMAT ENCOUNTERED WHILE READING FILE \*\*\*\*.

File is not in the correct format. Either the wrong tape was mounted or it does not contain what you think it should.

601 \*\*\* USER FATAL MESSAGE 601, THE KEYWORD ON THE ABOVE CARD IS ILLEGAL OR MISSPELLED. SEE THE FOLLOWING LIST FOR LEGAL KEYWORDS.

Case control expects each card to begin with a keyword (usually 4 characters in length). Your card does not. User message 612 will list the legal keywords along with a brief description of function. To remove the error, consult message 612 or NASTRAN case control card descriptions and spell your request correctly.

602 \*\*\* USER WARNING MESSAGE 602, TWO OR MORE OF THE ABOVE CARD TYPES DETECTED WHERE ONLY ONE IS LEGAL. THE LAST FOUND WILL BE USED.

Remove the card with the duplicate meaning. Note that some cards have alternate forms.

603 \*\*\* USER FATAL MESSAGE 603, THE ABOVE CARD DOES NOT END PROPERLY. COMMENTS SHOULD BE PRECEDED BY A DOLLAR SIGN.

Case control cards of the form name = value should not contain more than one value. Consult your NASTRAN case control deck document for a complete description of the card or precede your comments with a dollar sign.

604 \*\*\* USER FATAL MESSAGE 604, THE ABOVE CARD HAS A NONINTEGER IN AN INTEGER FIELD.

Consult your NASTRAN case control deck document for legal values.

608 \*\*\* USER FATAL MESSAGE 608, THE SET ID SPECIFIED ON THE ABOVE CARD MUST BE DEFINED PRIOR TO THIS CARD.

Set identification numbers must be specified prior to their use. Also sets specified within a subcase die at the end of the subcase. Redefine set (or define set) or move set out of subcase.

609 \*\*\* USER FATAL MESSAGE 609, SUBCASE DELIMITER CARDS MUST HAVE A UNIQUE IDENTIFYING INTEGER.

Subcase type cards must have an identifying integer. These numbers must be strictly increasing. Renumber your subcase cards. The use of a nonblank delimiter (e.g., =) will also cause this message to occur.

610 \*\*\* USER FATAL MESSAGE 610, THE VALUE FOLLOWING THE EQUAL SIGN IS ILLEGAL.

Case control cannot identify the BCD value after the equal sign. Consult NASTRAN case control card descriptions for a full description of the card.

611 \*\*\* USER FATAL MESSAGE 611, TEN CARDS HAVE ILLEGAL KEYWORDS. NASTRAN ASSUMES BEGIN BULK CARD IS MISSING. IT WILL NOW PROCESS YOUR BULK DATA.

Only ten keywords may be misspelled. A common source of this error may be the omission of the OUTPUT(PLOT) delimiter cards.

612 \*\*\* USER FATAL MESSAGE 612, --LIST OF LEGAL CASE CONTROL MNEMONICS.

This message is caused by messages 601 or 611.

613 \*\*\* USER FATAL MESSAGE 613, THE ABOVE SET CONTAINS 'EXCEPT' WHICH IS NOT PRECEDED BY 'THRU'.

Only identification numbers included in THRU statements may be excepted. Simplify your SET request.

614 \*\*\* USER MESSAGE 614, THE ABOVE SET IS BADLY SPECIFIED.

The grammar of the SET list is so confused that IFPL cannot continue. Simplify the SET list.

615 \*\*\* USER FATAL MESSAGE 615, AN IMPROPER OR NO NAME GIVEN TO THE ABOVE SET.

SET lists must have integer names. This SET list does not have one. SET 10 = is the correct format. Give the SET a correct integer name.

616 \*\*\* USER FATAL MESSAGE 616, 'EXCEPT' CANNOT BE FOLLOWED BY 'THRU'. LIST EXPLICITLY ALL EXCEPTIONS.

EXCEPT in SET list can only be followed by integers. An integer larger than the THRU pair terminates THRU. Either list exceptions explicitly, use two 'THRU's, or terminate first THRU.

617 \*\*\* USER FATAL MESSAGE 617, A NONPOSITIVE INTEGER APPEARS IN A POSITIVE POSITION.

Most integer values in case control must be positive. The above card either has a negative integer or a BCD value in a positive position. Check the case control deck documentation for the proper card format.

618 \*\*\* USER FATAL MESSAGE 618, PLOTTER OUTPUT IS REQUESTED BUT NO PLOT TAPE IS SET UP.

Neither PLT1 or PLT2 is a physical tape. Remove the plot control packet or set up the appropriate tape.

619 \*\*\* USER WARNING MESSAGE 619, SET MEMBER \*\*\* BELONGS TO \*\*\* THRU \*\*\*.

A set member is already included in a THRU. The individual member will be absorbed in the THRU.

620 \*\*\* USER WARNING MESSAGE 620, DUPLICATE \*\*\* IS IN SET LIST.

A set member is listed twice. The second reference will be deleted.

625 \*\*\* USER FATAL MESSAGE 625, SUBCASE ID'S MUST BE LESS THAN 99,999,999.

Reduce the size of your subcase identification number. Note also that BCD subcase identification numbers are not legal.

627 \*\*\* USER FATAL MESSAGE 627, THE ABOVE SUBCASE HAS BOTH A STATIC LOAD AND A REAL EIGENVALUE METHOD SELECTION -- REMOVE ONE.

The buckling problem requires two subcases: one for statics and one for buckling. Both a load and a method selection cannot take place in the same subcase.

997 \*\*\* USER WARNING MESSAGE 997, NO. \*\*\*. FRAME NO. \*\*\*\* INPUT DATA INCOMPATIBLE. ASSUMPTIONS MAY PRODUCE INVALID PLOT.

NO. \*\*\* may take any value from 1 to 4 with the following meaning:

1. Specified X maximum equals X minimum. If this value is zero, then X maximum is set to 5.0 and X minimum to -5.0; otherwise, 5 times the absolute value of X maximum is added to X maximum and subtracted from X minimum.
2. Specified X maximum is smaller than X minimum. The values are reversed.
3. Same meaning as number 1 for Y maximum and Y minimum.
4. Same meaning as number 2 for Y maximum and Y minimum.

### 6.3 Functional Module Messages

Functional module messages indicate that a problem exists during the execution of a particular module.

2002 \*\*\* SYSTEM FATAL MESSAGE 2002, GRID POINT \*\*\*\* NOT IN EQEXIN.

This message indicates a problem design error in GPL.

2003 \*\*\* USER FATAL MESSAGE 2003, COORDINATE SYSTEM \*\*\*\* REFERENCES UNDEFINED GRID POINT \*\*\*\*.

Applies to COORDj definitions.

2005 \*\*\* SYSTEM FATAL MESSAGE 2005, INCONSISTENT COORDINATE SYSTEM DEFINITION.

At least one coordinate system cannot be tied to the basic system.

2006 \*\*\* USER FATAL MESSAGE 2006, INTERNAL GRID POINT \*\*\*\* REFERENCES UNDEFINED COORDINATE SYSTEM \*\*\*\*.

The grid point whose internal sequence number is printed above references an undefined coordinate system in either field 3 or field 7 of a GRID card.

2013 \*\*\* USER WARNING MESSAGE 2013, NO STRUCTURAL ELEMENTS EXIST.

Model checked for structural elements.

2015 \*\*\* USER WARNING MESSAGE 2015, EITHER NO ELEMENTS CONNECT INTERNAL GRID POINT \*\*\*\*\* OR IT IS CONNECTED TO A RIGID ELEMENT OR A GENERAL ELEMENT.

The message is a warning only since the degrees of freedom associated with the point may be removed. The internal identification number is formed by assigning to each grid point one of the integers 1 or 2, depending on its resequenced position.

2021 \*\*\* SYSTEM FATAL MESSAGE 2021, BAD GMMAT CALLING SEQUENCE.

The calling sequence of the subroutine which calls either subroutine GMMATD or GMMATS defined a nonconformable matrix product. The subroutine examines the transpose flags in combination with the orders of the matrices to make sure that a conformable matrix product is defined by this input data. This test clearly is made for purposes of calling routine check-out only. No tests are made, nor can they be made to insure that the calling routine has provided sufficient storage for arrays.

2023 \*\*\* SYSTEM FATAL MESSAGE 2023, DETCK UNABLE TO FIND PIVOT POINT \*\*\*\* IN GPCT.

Probable error in creating the ECPT data block in module TAL.

2024 \*\*\* USER FATAL MESSAGE 2024, OPERATION CODE \*\*\*\*\* NOT DEFINED FOR MODULE PARAM.

The use of V,N,SUB rather than C,N,SUB can cause this.

2025 \*\*\* USER FATAL MESSAGE 2025, UNDEFINED ~~C~~ORDINATE SYSTEM \*\*\*\*.

The coordinate system identification number transmitted by means of ECPT(1) could not be found in the CSTM array. The user should check coordinate system numbers used on bulk data cards against those defined on ~~C~~ORD1C, ~~C~~ORD1R, or ~~C~~ORD1S bulk data cards to insure that there are no undefined coordinate systems.

2026 \*\*\* USER FATAL MESSAGE 2026, ELEMENT \*\*\*\* ~~G~~OMETRY YIELDS UNREASONABLE MATRIX.

Referenced element geometry and/or properties yield a numerical result which causes an element stiffness or mass matrix to be undefined. Possible causes include, but are not limited to the fact that the length of a bar is zero because the end points have the same coordinates, that the sides of a triangle are colinear, which leads to a zero cross product in defining an element coordinate system, or that the bar orientation vector is parallel to the bar axis. Check GRID bulk data cards defining element end points for bad data.

2034 \*\*\* SYSTEM FATAL MESSAGE 2034, ELEMENT \*\*\*\* SIL'S ~~D~~ ~~N~~OT MATCH PIV~~O~~T.

Possible error in generation of the ECPT data block.

2042 \*\*\* USER FATAL MESSAGE 2042, MISSING MATERIAL TABLE \*\*\*\* ~~F~~OR ELEMENT \*\*\*\*.

The referenced material table identification number is missing. The user should check to see that all element property bulk data cards (e.g., PBAR) reference material card identification numbers for material property cards that exist in the bulk data deck.

2049 \*\*\* USER FATAL MESSAGE 2049, UNDEFINED GRID P~~O~~INT \*\*\*\* HAS AN ~~O~~MITTED ~~C~~ORDINATE.

An ~~O~~MIT card references a grid point which has not been defined.

2050 \*\*\* USER FATAL MESSAGE 2050, UNDEFINED GRID P~~O~~INT \*\*\*\* HAS A SUP~~ORT ~~C~~ORDINATE.~~

A SUP~~ORT card references a grid point which has not been defined.~~

2052 \*\*\* USER FATAL MESSAGE 2052, UNDEFINED GRID P~~O~~INT \*\*\* IN SINGLE-P~~O~~INT ~~C~~ONSTRAINT SET \*\*\*\*.

An SPC card in the selected SPC set references a grid point which has not been defined.

2053 \*\*\* USER FATAL MESSAGE 2053, UNDEFINED SINGLE-P~~O~~INT ~~C~~ONSTRAINT SET \*\*\*\*.

A single-point constraint set selected in the case control deck could not be found on either an SPCADD card, or SPC card; or a set referenced on an SPCADD card could not be found on an SPC card.

2072 \*\*\* SYSTEM WARNING MESSAGE 2072, CARD TYPE \*\*\* NOT FOUND ON DATA BLOCK.

This warning message is issued when the trailer bit for the card type equals 1, but the corresponding record is not on the data block.

2081 \*\*\* USER FATAL MESSAGE 2081, NULL DIFFERENTIAL STIFFNESS MATRIX.

Differential stiffness is not defined for all structural elements. Only TRMEM, TRIA2, and BAR are defined for differential stiffness calculations. The user has not included any of these elements in his model and therefore a null differential stiffness matrix was generated.

2083 \*\*\* USER FATAL MESSAGE 2083, NULL DISPLACEMENT VECTOR.

The displacement vector for the linear solution part of a static analysis with differential stiffness problem is the zero vector. Check loading conditions.

2085 \*\*\* USER INFORMATION MESSAGE 2085, \*\*\*\* SPILL, NPVT \*\*\*\*.

During processing of the ECPT data block in module \*\*\*\*, so many elements were attached to the referenced pivot point (NPVT) that module spill logic was initiated.

2086 \*\*\* USER INFORMATION MESSAGE 2086, SMA2 SPILL, NPVT \*\*\*\*.

See explanation for message 2085.

2088 \*\*\* USER FATAL MESSAGE 2088, DUPLICATE TABLE ID \*\*\*\*.

All tables must have unique numbers. Check for uniqueness.

2089 \*\*\* USER FATAL MESSAGE 2089, TABLE \*\*\*\* UNDEFINED.

The table number in the list of table numbers input to subroutine PRETAB by means of argument 7 was not found after reading the DIT data block. Check list of tables in the bulk data deck.

2090 \*\*\* SYSTEM FATAL MESSAGE 2090, TABLE DICTIONARY ENTRY \*\*\*\* MISSING.

Logic error in subroutine PRETAB, or open core used by PRETAB has been destroyed.

2101A \*\*\* USER FATAL MESSAGE 2101A, GRID POINT \*\*\*\* COMPONENT \*\*\* ILLEGALLY DEFINED IN SETS \*\*\*\*.

The above grid point and component have been defined in each of the above dependent subsets. A point may belong to, at most, one dependent subset.

2102 \*\*\* USER WARNING MESSAGE 2102, LEFT-HAND MATRIX ROW POSITION \*\*\*\* OUT OF RANGE - IGNORED.

A term in the A matrix whose row position is larger than the stated dimension was detected and ignored.

2103 \*\*\* SYSTEM FATAL MESSAGE 2103, SUBROUTINE MAT WAS CALLED WITH INFLAG=2, THE SINE OF ANGLE X, MATERIAL ORIENTATION ANGLE, NONZERO, BUT  $\sin^2(X) + \cos^2(X)$  DIFFERED FROM 1 IN ABSOLUTE VALUE BY MORE THAN .0001.

A check is made in MAT to insure that  $\text{ABS}(\sin(\text{THETA})^2 + \cos(\text{THETA})^2 - 1.00) \leq .0001$  when INFLAG equals 2. The calling routine did not set SINTH and COSTH cells in /MATIN/ properly.

2104 \*\*\* USER FATAL MESSAGE 2104, UNDEFINED COORDINATE SYSTEM \*\*\*\*.

See the explanation for message 2025.

2107 \*\*\* USER FATAL MESSAGE 2107, EIG-CARD FROM SET \*\*\*\* REFERENCES DEPENDENT COORDINATE OR GRID POINT \*\*\*\*.

When the point option is used on an EIGB or EIGR card, the referenced point and component must be in the analysis set for use in normalization.

2111 \*\*\* USER WARNING MESSAGE 2111, BAR \*\*\*\* COUPLED BENDING INERTIA SET TO 0.0 IN DIFFERENTIAL STIFFNESS.

The coupled bending inertia term on a PBAR card, if nonzero, is set to zero in the differential stiffness routine for the BAR.

2112 \*\*\* SYSTEM FATAL MESSAGE 2112, UNDEFINED TABLE \*\*\*\*.

The referenced table number could not be found in core.

2116 \*\*\* SYSTEM FATAL MESSAGE 2116, MATID \*\*\*\* TABLEID \*\*\*\*.

The referenced material table identification number could not be found among the set of all MATI cards in core.

2132 \*\*\* USER FATAL MESSAGE 2132, NON-ZERO SINGLE POINT CONSTRAINT VALUE SPECIFIED BUT DATA BLOCK YS IS PURGED.

Many DMAP sequences do not support constrained displacements (especially dynamic solutions). An attempt to specify a constrained displacement in these cases results in this message.

2137 \*\*\* USER FATAL MESSAGE 2137, PROGRAM RESTRICTION FOR MODULE SSG1 - ONLY 100 LOAD SET ID'S ALLOWED. DATA CONTAINS \*\*\*\* LOAD SET ID'S.

Reduce the number of load set ID's.

2138 \*\*\* USER FATAL MESSAGE 2138, ELEMENT IDENTIFICATION NUMBER \*\*\*\* IS TOO LARGE.

Element identification numbers on connection cards must be less than 16,777,215.

2139 \*\*\* USER FATAL MESSAGE 2139, ELEMENT \*\*\*\* IN DEFORM SET \*\*\*\* IS UNDEFINED.

A selected element deformation set includes an element twice, a non-existent element, or a non-one-dimensional element.

2140 \*\*\* USER FATAL MESSAGE 2140, GRID POINT OR SCALAR POINT ID \*\*\*\* IS TOO LARGE.

Program restriction on the size of integer numbers. A card defining a grid point or scalar point has a number larger than 2,000,000.

2264 \*\*\* SYSTEM FATAL MESSAGE 2264, NUMBER OF ROWS COMPUTED (\*\*\*\*) WAS GREATER THAN SIZE REQUESTED FOR OUTPUT MATRIX (\*\*\*\*).

Module ADD determines size of output matrices (j set size). Sum of number of rows added by different method totals more than maximum allowed.

2288 \*\*\* SYSTEM FATAL MESSAGE 2288, \*\*\*\* READ INCORRECT NUMBER WORDS (\*\*\*\* \*\*\*\*).

Subroutine \*\*\*\* read \*\*\*\* words on the \*\*\*\* card which is incorrect.

2296 \*\*\* USER FATAL MESSAGE 2296, INSUFFICIENT CORE \*\*\* (\*\*\*\*), ELEMENT \*\*\*\*.

Subroutine \*\*\*\* has insufficient core when loading element type of number \*\*\*\*. Elements are read into core by element type (see /GPTAL/ sequence) and then by sequential element number.

2297 \*\*\* SYSTEM FATAL MESSAGE 2297, INCORRECT LOGIC FOR ELEMENT TYPE \*\*\*\*, ELEMENT \*\*\*\*, (\*\*\*\*).

Subroutine (\*\*\*\*) has sequential element search. Element type can be found in /GPTAL/.

2298 \*\*\* USER FATAL MESSAGE 2298, INSUFFICIENT CORE \*\*\*\* (\*\*\*\*), PROPERTY \*\*\*\*.

Subroutine \*\*\*\* (core \*\*\*\*) had insufficient core when loading property \*\*\*\*.

3001 \*\*\* SYSTEM FATAL MESSAGE 3001, ATTEMPT TO OPEN DATA SET \*\*\* IN SUBROUTINE \*\*\*\*\* WHICH WAS NOT DEFINED IN FIST.

Subroutine did not expect data block to be purged. Check data block requirements for module.



3002 \*\*\* SYSTEM FATAL MESSAGE 3002, EOF ENCOUNTERED WHILE READING DATA SET  
\*\*\*\*\* (FILE \*\*\*) IN SUBROUTINE \*\*\*\*\*.

This message is issued when an end of file occurs while trying to skip the header record. The data block is not in the proper format.

3003 \*\*\* SYSTEM FATAL MESSAGE 3003, ATTEMPT TO READ PAST THE END OF A LOGICAL  
RECORD IN DATA SET \*\*\*\*\* (FILE \*\*\*) IN SUBROUTINE \*\*\*\*\*.

This message is issued when the file is positioned at the beginning of a logical record and the record does not contain at least three words. Data block is not in proper format.

3005 \*\*\* USER FATAL MESSAGE 3005, ATTEMPT TO OPERATE ON SINGULAR MATRIX \*\*\*\*  
IN SUBROUTINE \*\*\*\*.

A diagonal term does not exist for a column of (U). This is normally detected in DECAMP and implies care was not taken in processing singular matrices in the calling routine.

3009 \*\*\* SYSTEM FATAL MESSAGE 3009, DATA TRANSMISSION ERROR ON DATA SET  
\*\*\*\*\* (FILE \*\*\*) .

Remove the PLOT request from the NASTRAN job.

3010 \*\*\* SYSTEM FATAL MESSAGE 3010, ATTEMPT TO MANIPULATE DATA SET \*\*\*\*\*  
(FILE \*\*\*) BEFORE OPENING FILE.

An operation other than OPEN or CLOSE is requested on a file which is not defined in the FIST.

3011 \*\*\* SYSTEM FATAL MESSAGE 3011, ATTEMPT TO WRITE A TRAILER ON FILE \*\*\*  
WHEN IT HAS BEEN PURGED.

The file did not exist in the FIST when WRTTRL was called.

3012 \*\*\* SYSTEM FATAL MESSAGE 3012, ATTEMPT TO OPEN DATA SET \*\*\*\*\* (FILE \*\*\*)  
WHICH HAS ALREADY BEEN OPENED.

GIN OPEN was called while the file was already open.

3013 \*\*\* SYSTEM FATAL MESSAGE 3013, ATTEMPT TO READ DATA SET \*\*\*\*\* (FILE \*\*\*)  
WHEN IT WAS OPENED FOR OUTPUT.

GIN was called to READ a data block opened for output.

3014 \*\*\* SYSTEM FATAL MESSAGE 3014, ATTEMPT TO WRITE DATA SET \*\*\*\*\* (FILE \*\*\*)  
WHEN IT WAS OPENED FOR INPUT.

GIN was called to WRITE a data block opened for input.

3015 \*\*\* SYSTEM FATAL MESSAGE 3015, ATTEMPT TO FWDREC ON DATA SET \*\*\*\*\*  
(FILE \*\*\*) WHEN IT WAS OPENED FOR OUTPUT.

GINØ was called to FWDREC a file opened for output.

3016 \*\*\* SYSTEM FATAL MESSAGE 3016, \*\*\*\* MATRIX IS NOT IN PROPER FORM IN  
SUBROUTINE \*\*\*\*.

This implies that the input matrix is not in the proper form or type acceptable to the subroutine. Check the trailer information on the matrix and the subroutine description for the discrepancy.

3017 \*\*\* USER WARNING MESSAGE 3017, ONE OR MORE GRID POINT SINGULARITIES  
HAVE NOT BEEN REMOVED BY SINGLE POINT CONSTRAINTS.

Singularities or near singularities may exist at the grid-point level. The listed singularities should be examined for data errors. The check performed here is neither necessary nor sufficient for a singular matrix.

3018 \*\*\* SYSTEM FATAL MESSAGE 3018, MODULE \*\*\*\*\*, SEQUENCE NO. \*\*\*,  
REQUIREMENTS EXCEED AVAILABLE FILES.

Segment file allocator (SFA) did not have sufficient logical files available to fill the request of the module. Cut module requirements or increase the logical files within the computer system.

3019 \*\*\* USER FATAL MESSAGE 3019, MAXIMUM LINE COUNT EXCEEDED IN SUBROUTINE \*\*\*\*  
LINE COUNT EQUALS \*\*\*\*.

The total number of lines written on the system output file has exceeded the set limit (default value is 20,000). If you wish to increase this value, include a card of the form "MAXLINES=n" in your case control deck.

3021 \*\*\* SYSTEM FATAL MESSAGE 3021, FILE \*\*\* NOT DEFINED IN FIST.

An operation other than OPEN or CLOSE is requested on a file which is not defined in the FIST.

3022 \*\*\* SYSTEM WARNING MESSAGE 3022, DATA SET \*\*\*\*\* IS REQUIRED AS INPUT  
AND IS NOT OUTPUT BY A PREVIOUS MODULE IN THE CURRENT DMAP ROUTE.

SFA detected that an input data block to a future module has not been generated. If the future module requires that this data block exist, the module may terminate with a fatal error. This message can occur (and most often does) when the SFA has removed from its tables (due to a need for more room) previously purged data blocks. In this case no error or even a warning is implied.

3023 \*\*\* USER INFORMATION MESSAGE 3023--PARAMETERS FOR SYMMETRIC  
DECOMPOSITION OF DATA BLOCK \*\*\*\*\* (N = \*\*\*\*\*) TIME  
ESTIMATE = \*\*\*\*\*.

C AVG = \*\*\*\*\*                    S AVG = \*\*\*\*\*                    PC MAX = \*\*\*\*\*  
PC AVG = \*\*\*\*\*                    ADDITIONAL CORE = \*\*\*\*\*                    PC GROUPS = \*\*\*\*\*  
SPILL GROUPS = \*\*\*\*\*                    C MAX = \*\*\*\*\*                    PREFACE LOOPS = \*\*\*\*\*

N is the number of rows in the data block; TIME is the estimate (in seconds) to perform the decomposition; C AVG is the average number of active columns per pivot row; PC AVG is the average number of passive columns at each active termination point; SPILL GROUPS is the number of spill groups; S AVG is the average number of rows in each spill group; ADDITIONAL CORE (positive) is the amount of core required to avoid spill, (negative) is the amount of unused core; C MAX is the maximum number of active columns in any one pivot row; PC MAX is the maximum number of passive columns at any one active column termination point; PC GROUPS is the number of active column termination points; PREFACE LOOPS is the number of times the preface of the decomposition subroutine is executed.

3024 \*\*\* USER INFORMATION MESSAGE 3024, THE BANDWIDTH OF MATRIX \*\*\*\* EXCEEDS  
THE MAXIMUM BANDWIDTH. A MAXIMUM BANDWIDTH OF \*\*\*\* WILL BE USED.

This message indicates that a matrix has scattered terms way off the diagonal (i.e., a large bandwidth). Instead of searching all combinations of B and C, the search is started at the maximum bandwidth.

3025 \*\*\* SYSTEM FATAL MESSAGE 3025, ILLEGAL INDEX IN ACTIVE ROW OR COLUMN  
CALCULATION IN \*\*\*\*.

Possible machine error. Rerun problem. If error persists, a code error exists in the decomposition routine.

3026 \*\*\* SYSTEM FATAL MESSAGE 3026, MATRIX \*\*\*\* EXCEEDS MAXIMUM ALLOWABLE  
SIZE FOR BANDWIDTH PLUS ACTIVE COLUMNS. BMAX = \*\*\*\*, CMAX = \*\*\*\*.

Sufficient space was not reserved for the generation of the B versus C vector. SDCOMP should be recompiled to increase BMAX and CMAX.

3027 \*\*\* USER INFORMATION MESSAGE 3027, \*\*\*\* DECOMPOSITION TIME ESTIMATE  
IS \*\*\*\*\* SECONDS.

Gives the estimated time required for a decomposition in seconds and the type of matrix (i.e., complex, real (double or single precision), symmetric, or unsymmetric).

3028 \*\*\* USER INFORMATION MESSAGE 3028, B = \*\*\*\*, BBAR = \*\*\*\*, C = \*\*\*\*, CBAR = \*\*\*\*, R = \*\*\*\*.

Gives the upper bandwidth (B), lower bandwidth (BBAR), number of active columns (C), and active rows (CBAR) used in the unsymmetric decomposition.

3029 \*\*\* SYSTEM FATAL MESSAGE 3029, PHYSICAL END-OF-FILE ENCOUNTERED ON DATA SET \*\*\*\* (FILE \*\*\*\*).

Since logical end of files are used by GINØ, a physical end of file indicates an attempt to read beyond valid data.

3032 \*\*\* USER FATAL MESSAGE 3032, UNABLE TO FIND SELECTED SET (\*\*\*\*) IN TABLE (\*\*\*\*) IN SUBROUTINE.

A particular set used in the problem was not included in the data. Good examples are loads, initial conditions, or frequency sets. Include the required data or change the case control deck to select data already in problem. Set zero has a special meaning. A set selection was required, but none was made. For example, no METHODØ was selected for an eigenvalue extraction problem.

This message can also indicate that a LOAD card has referenced another LOAD card, which is not permitted.

3034 \*\*\* USER WARNING MESSAGE 3034, ORTHOGONALITY CHECK FAILED, LARGEST TERM = \*\*\*\* EPSI = \*\*\*\*.

The off-diagonal terms of the modal mass matrix are larger than the user input criteria on the EIGB or EIGR bulk data card. The eigenvectors are not orthogonal to this extent. This nonorthogonality is especially important if a modal formulation is contemplated.

3035 \*\*\* USER INFORMATION MESSAGE 3035, FOR LOAD \*\* EPSILON SUB E = \*\*\*\*\*.

This is an informative message reflecting the accumulated round-off error of the static solution.

3037 \*\*\* SYSTEM FATAL MESSAGE 3037, JOB TERMINATED IN SUBROUTINE \*\*\*\*.

This message designates the subroutine in which the program terminated. It should be preceded by a user message which explains the cause of the termination. The module in which the program terminated can be found by examining the on-line time messages.

3038 \*\*\* SYSTEM FATAL MESSAGE 3038, DATA SET \*\*\* DOES NOT HAVE MULTIREEL CAPABILITY.

Computer hardware or software does not support multireel files.

3039 \*\*\* SYSTEM FATAL MESSAGE 3039, ENDSYS CANNOT FIND SAVE FILE.

File cannot be found to save and restore executive tables during link switching.

3040 \*\*\* SYSTEM FATAL MESSAGE 3040, ATTEMPT TO WRITE DATA SET \*\*\*\*\* (FILE \*\*\*)  
WHEN IT IS AN INPUT FILE.

Input data blocks for a module (100 .LT. NAME .LT. 200) may be read only.

3045 \*\*\* USER WARNING MESSAGE 3045, INSUFFICIENT TIME TO COMPLETE THE  
REMAINING \*\* SOLUTION(S) IN MODULE \*\*\*.

The time specified on the NASTRAN TIME card has expired in the named module. The module will be terminated. NASTRAN will continue running until the time on the job card expires. Rerun problem with increased time.

3048 \*\*\* SYSTEM FATAL MESSAGE 3048, BUFFER CONTROL WORD INCORRECT FOR  
GINØ \*\*\*\* OPERATION ON DATA BLOCK \*\*\*\*.

The buffer control word has been destroyed outside of GINØ or an attempt to READ a file opened to WRITE or similar error has occurred.

3049 \*\*\* SYSTEM FATAL MESSAGE 3049, GINØ UNABLE TO POSITION DATA BLOCK \*\*\*\*  
CORRECTLY DURING \*\*\*\* OPERATION.

A block number read does not match the expected block number. The file has been repositioned outside the GINØ environment or a machine or operating system error has occurred.

3050 \*\*\* USER FATAL MESSAGE 3050, INSUFFICIENT TIME REMAINING FOR  
DECOMPOSITION, \*\*\*\*. TIME ESTIMATE IS \*\*\*\* SECONDS.

The time estimated for a decomposition exceeds the remaining time. Increase the time estimate for the run.

3054 \*\*\* USER WARNING MESSAGE 3054, THE ACCURACY OF EIGENVECTOR \*\*\*\*  
CORRESPONDING TO THE EIGENVALUE \*\*\*\* IS IN DOUBT.

The eigenvector failed to converge in the allowable number of iterations. Particular attention should be given to the off-diagonal terms of the modal mass matrix (MI) to determine if this vector is orthogonal to the remaining vectors. These terms will be computed and checked if field 9 on the EIGR card contains a nonzero value. The message is expected in the case of close or multiple eigenvalues even though the vectors are properly computed.

3055 \*\*\* USER FATAL MESSAGE 3055, AN ATTEMPT TO MULTIPLY OR MULTIPLY AND ADD NON-CONFORMABLE MATRICES TOGETHER WAS MADE IN MODULE \*\*\*\*\*.

The multiply/add subroutine requires conformable matrices. There are two possible equations:

1.  $[X] = [A][B] + [C]$

The number of columns of  $[A]$  must be equal to the number of rows of  $[B]$ ; the number of columns of  $[C]$  must be equal to the number of columns of  $[B]$ ; and the number of rows of  $[C]$  must be equal to the number of rows of  $[A]$ .

2.  $[X] = [A]^T[B] + [C]$

The number of rows of  $[A]$  must be equal to the number of rows of  $[B]$ ; the number of columns of  $[C]$  must be equal to the number of columns of  $[B]$ ; and the number of rows of  $[C]$  must be equal to the number of columns of  $[A]$ .

3056 \*\*\* USER FATAL MESSAGE 3056, NO MASS MATRIX IS PRESENT BUT MASS DATA IS REQUIRED.

An operation with the mass matrix is required, but none was created. A typical cause is the omission of RH0 on the MAT1 card.

3057 \*\*\* USER FATAL MESSAGE 3057, MATRIX \*\*\*\* IS NOT POSITIVE DEFINITE.

A Cholesky decomposition was attempted on the above matrix, but a diagonal term was negative or equal to zero such that the decomposition failed.

3058 \*\*\* USER WARNING MESSAGE 3058, EPSILON IS LARGER THAN \*\*\*\* FOR SUBCASE \*\*\*\*.

The error residual (either  $\epsilon_Q$  or  $\epsilon_\phi$ )

$$\epsilon = \frac{\{u\}^T \{\delta P\}}{\{P\}^T \{u\}} \text{ is larger than would be expected for a well-}$$

conditioned problem. Near singularities may exist.

3082 \*\*\* USER WARNING MESSAGE 3082, M = \*\*\*\*\*, N = \*\*\*\*\*.

More than one n-set degree of freedom is associated with an m-set degree of freedom. The term associated with the m-n indices given in the message is ignored.

3083 \*\*\* USER FATAL MESSAGE 3083, UM POSITION = \*\*\*\*\*, SIL = \*\*\*\*\*.

An m-set degree of freedom is not expressed in terms of an n-set degree of freedom.

3093 \*\*\* SYSTEM FATAL MESSAGE 3093, ELEMENT = \*\*\*\*\* REASON = \*\*\*\*\*.

1. Less than 2 points have been referenced.
2. Unable to locate SIL value.
3. Unrecognized form for element.
4. Illegal number of points for this form of the element.

3097 \*\*\* USER FATAL MESSAGE 3097, COLUMN \*\*\*\*\* IS SINGULAR. SYMMETRIC \*\*\*\*\* DECOMP.

When a matrix being read in is singular (null column or, for symmetric decomposition, a zero diagonal), the internal column number and type of decomposition are identified. The message does not appear for special cases, such as less than three columns, or for proportional rows.

3131 \*\*\* USER FATAL MESSAGE 3131, INPUT STIFFNESS AND MASS MATRICES ARE NOT COMPATIBLE.

The matrices must be the same size to properly perform matrix operations.

4028 \*\*\* USER FATAL MESSAGE 4028, THE CODE FOR GRID POINT \*\*\*\*\* DOES NOT MATCH THE CODE FOR GRID POINT \*\*\*\*\*.

A GRID point on SIDE 1 must be connected to a GRID point on SIDE 2 and a SCALAR point on SIDE 1 must be connected to a SCALAR point on SIDE 2.

4066 \*\*\* USER FATAL MESSAGE 4066, SECOND OUTPUT DATA BLOCK MUST NOT BE PURGED.

The transformation matrix between physical and symmetric components does not exist. Insure that the number of case control subcases is specified correctly and that the component loads are properly ordered.

4090 \*\*\* USER FATAL MESSAGE 4090, ONE OF THE FOLLOWING NON-ZERO IDENTIFICATION NUMBERS APPEARS ON SOME COMBINATION GRID BULK DATA CARDS. ID = XXX, ID = XXX, ID = XXX.

All GRID data cards should have unique identification numbers.

6007 \*\*\* SYSTEM FATAL MESSAGE 6007, IMPROPER FILE SETUP FOR \*\*\*.

An external input/output operation has been defined, but the file is missing or the card is improper.

6010 \*\*\* SYSTEM FATAL MESSAGE 6010, ILLEGAL VARIABLE TO BE SET IN DMAP STATEMENT, (N).

The system has encountered an illegal type of word to be inserted in a DMAP sequence. Could possibly occur if a floating point number were used instead of an integer on an input card.

6015 \*\*\* USER FATAL MESSAGE 6015, TOO MANY CHARACTERS TO BE INSERTED IN A DMAP LINE. N=\*\*\*.

A BCD word has been defined with too many characters to fit the space in the DMAP. (Usual limit is eight.) Message could also occur if block data subprogram ASDBD has an error.

6016 \*\*\* USER FATAL MESSAGE 6016, TOO MANY DIGITS TO BE INSERTED IN DMAP VALUE= \*\*\*.

An integer is limited to eight digits.

#### 6.4 Analysis Error Messages

A number of fatal errors are detected by DMAP statements. These messages indicate the presence of fatal user errors that either cannot be determined by the functional modules or that can be more effectively detected by DMAP statements. The detection of such an error causes a transfer to a LABEL instruction near the end of the rigid format. The text of the message is output and the execution is terminated. These messages will always appear at the end of the NASTRAN output.

##### 6.4.1 Error Messages for Static Analysis

NØ. 1 - ATTEMPT TO EXECUTE MORE THAN 100 LOOPS.

An attempt has been made to use more than 100 different sets of boundary conditions. This number may be increased by altering the REPT instruction following SDRI.

NØ. 2 - MASS MATRIX REQUIRED FOR WEIGHT AND BALANCE CALCULATIONS.

The mass matrix is null because either no elements were defined with connection cards, nonstructural mass was not defined on a property card, or the density was not defined on a material card.



NO. 3 - NO INDEPENDENT DEGREES OF FREEDOM HAVE BEEN DEFINED.

Either no degrees of freedom have been defined on GRID, SPPOINT, or scalar connection cards or all defined degrees of freedom have been constrained by SPC, SUPPORT, or OMIT cards.

NO. 4 - NO ELEMENTS HAVE BEEN DEFINED.

The stiffness matrix is null because no elements have been defined on connection cards.

NO. 5 - A LOOPING PROBLEM RUN ON NON-LOOPING SUBSET.

A problem requiring boundary condition changes was run on subsets 1 or 3. The problem should be restarted on subset 0.

#### 6.4.2 Error Messages for Vibration Analysis

NO. 1 - MASS MATRIX REQUIRED FOR REAL EIGENVALUE ANALYSIS.

The mass matrix is null because either no structural elements were defined with connection cards, nonstructural mass was not defined on a property card, or the density was not defined on a material card.

NO. 2 - EIGENVALUE EXTRACTION DATA REQUIRED FOR REAL EIGENVALUE ANALYSIS.

Eigenvalue extraction data must be supplied on an EIGR card and METHOD must select an EIGR set in the case control deck.

NO. 3 - NO INDEPENDENT DEGREES OF FREEDOM HAVE BEEN DEFINED.

No degrees of freedom have been defined on GRID cards or all defined degrees of freedom have been constrained by SPC, SUPPORT, or OMIT cards.

#### 6.4.3 Error Messages for Buckling Analysis

NO. 1 - NO STRUCTURAL ELEMENTS HAVE BEEN DEFINED.

The differential stiffness matrix is null because no structural elements have been defined with connection cards.

NO. 2 - FREE BODY SUPPORTS NOT ALLOWED.

Free bodies are not allowed in buckling analysis. The SUPPORT cards must be removed from the bulk data deck and other constraints applied if required for stability.

**NØ. 3 - EIGENVALUE EXTRACTION DATA REQUIRED FOR REAL EIGENVALUE ANALYSIS.**

Eigenvalue extraction data must be supplied on an EIGB card and METHOD must select an EIGB set in the case control deck.

**NØ. 4 - NO EIGENVALUES FOUND.**

No buckling modes exist in the range specified by the user.

**NØ. 5 - MASS MATRIX REQUIRED FOR WEIGHT AND BALANCE CALCULATIONS.**

The mass matrix is null because either no elements were defined with connection cards, nonstructural mass was not defined on a property card, or the density was not defined on a material card.

**NØ. 6 - NO INDEPENDENT DEGREES OF FREEDOM HAVE BEEN DEFINED.**

No degrees of freedom have been defined on GRID cards or all defined degrees of freedom have been constrained by SPC or OMIT cards.

## APPENDIX

### SAMPLE PROBLEMS

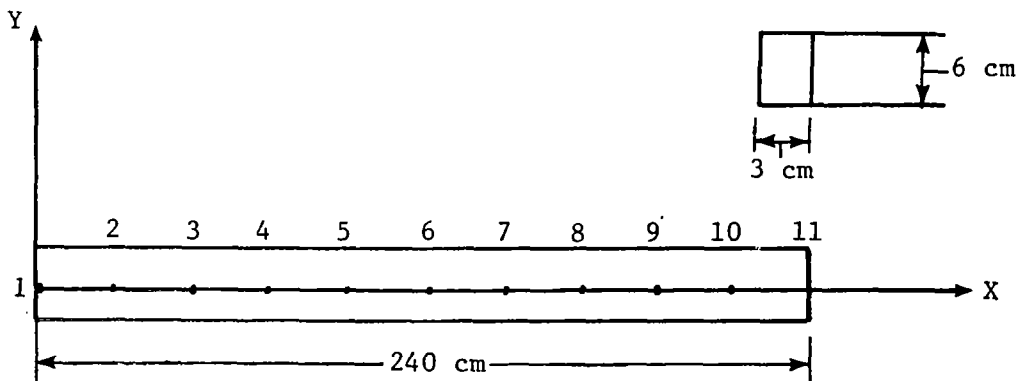
This appendix describes how to use a DMAP sequence to perform three types of analysis and presents a demonstration problem for matrix operations and data block manipulation. The three types of analysis are static analysis, vibration analysis, and buckling analysis. Each of the four problems is divided into the model, the DMAP sequence, a description of the major steps in the sequence, and a sample problem with output. The fourth problem has no model. To save space, the output of each problem has been compacted so that two to four pages of normal output are printed on one page in this guide.

The output for load vectors, displacement vectors, eigenvectors, and forces of SPC are under the headings T1, T2, T3, R1, R2, and R3. These headings are equivalent to displacement at X, Y, and Z and rotation about X, Y, and Z. The stresses output for the BAR element are under the headings SA1, SB1, SA2, SB2, SA3, SB3, SA4, and SB4. These symbols refer to the stress recovery coefficients on the PBAR card, where the subscript 1 is associated with the distance of a stress recovery point from plane 2 and the subscript 2 is associated with the distance from plane 1. The eigenvalue output is straightforward to read.

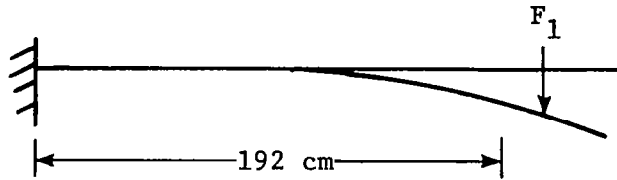
#### Static Analysis

This problem is a static analysis of the deflection of a uniform beam.

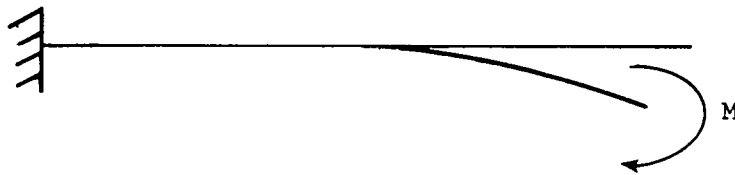
#### Model



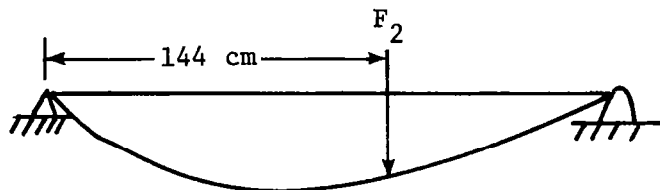
APPENDIX



Subcase 1.- Cantilever beam subjected to concentrated force.



Subcase 2.- Cantilever beam under moment of free end.



Subcase 3.- Simply supported beam with concentrated force.

## APPENDIX

### Static Analysis DMAP Sequence

```

1. XDMAP      ERR=2,LIST,REF $   STATIC ANALYSIS
2. FILE      OPTP2=SAVE/EST1=SAVE $
3. FILE      QG=APPEND/PGG=APPEND/UGV=APPEND/GM=SAVE/KNN=SAVE $
4. GP1       GEOM1,GEOM2,/GPL,EQEXIN,GPDT,CSTM,BGPD,SIL/S,N,LUSET/  V,N,
             NOGPD/V,N,ALWAYS=-1 $
5. GP2       GEOM2,EQEXIN/ECT $
6. PARAML    PCDB//C,N,PRES/C,N,/C,N,/C,N,/V,N,NOPCDB $
7. PURGE     PLTSETX,PLTPAR,GPSETS,ELSETS/NOPCDB $
8. COND      P1,NOPCDB $
9. PLTSET    PCDB,EQEXIN,ECT/PLTSETX,PLTPAR,GPSETS,ELSETS/S,N,NSIL/  S,N,
             JUMPPLOT=-1 $
10. PRTMSG   PLTSETX// $
11. PARAM    //C,N,MPY/V,N,PLTFLG/C,N,1/C,N,1 $
12. PARAM    //C,N,MPY/V,N,PFILE/C,N,0/C,N,0 $
13. COND     P1,JUMPPLOT $
14. PLOT     PLTPAR,GPSETS,ELSETS,CASECC,BGPD,EQEXIN,SIL,,,,/PLOTX1/  V,N,
             NSIL/V,N,LUSET/S,N,JUMPPLOT/S,N,PLTFLG/S,N,PFILE $
15. PRTMSG   PLOTX1// $
16. LABEL    P1 $
17. GP3      GEOM3,EQEXIN,GEOM2/SLT, /S,N,NOGRAV/V,N,NEVER=1 $
18. PARAM    //C,N,AND/V,N,NOMGG/V,N,NOGRAV/V,N,GRDPNT=-1 $
19. TA1      ECT,EPT,BGPD,SIL, ,CSTM/EST,GEI,GPECT,/V,N,LUSET/  S,N,
             NOSIMP/C,N,1/S,N,NOGENL/S,N,GENEL $
20. PARAM    //C,N,AND/V,N,NOELMT/V,N,NOGENL/V,N,NOSIMP $
21. COND     ERROR4,NOELMT $
22. PURGE    KGGX,GPST/NOSIMP/OGPST/GENEL $
23. PARAM    //C,N,MPY/V,N,CARDNO/C,N,0/C,N,0 $
24. COND     LBL1,NOSIMP $
25. PARAM    //C,N,ADD/V,N,NOKGGX/C,N,1/C,N,0 $
26. EQUIV    EST,EST1/NEVER $
27. EMG      EST,CSTM,MPT,DIT,GEOM2,/KELM,KDICT,MELM,MDICT,,/S,N,NOKGGX/ S,
             N,NOMGG/C,N,/C,N,/C,N,/C,N,/C,N,/C,N,COUPMASS/C,Y,CPBAR/C,Y,CPROD/C,Y,
             CPQUAD1/C,Y,CPQUAD2/C,Y,CPTRIA1/C,Y,CPTRIA2/ C,Y,CPTUBE/C,Y,
             CPQDPLT/C,Y,CPTRPLT/C,Y,CPTRBSC $
28. COND     JMPKGG,NOKGGX $
29. EMA      GPECT,KDICT,KELM/KNN ,GPST $
30. LABEL    JMPKGG $
31. COND     LBL1,NOMGG $
32. EMA      GPECT,MDICT,MELM/MGG,/C,N,-1/C,N,WTMASS=1.0 $
33. LABEL    LBL1 $
34. PARAM    //C,N,MPY/V,N,NSKIP/C,N,0/C,N,0 $
35. JUMP     LBL11 $
36. LABEL    LBL11 $
37. GP4      CASECC,GEOM4,EQEXIN,SIL,GPDT,BGPD,CSTM/RG,YS,USSET,ASET/V,N,
             LUSET/S,N,MPCF1/S,N,MPCF2/S,N,SINGLE/S,N,OMIT/S,N,REACT/S,N,
             NSKIP/S,N,REPEAT/S,N,NOSET/S,N,NOL/S,N,NDA/C,Y,SUBID $
38. COND     ERROR3,NOL $
39. PARAM    //C,N,AND/V,N,NOSR/V,N,SINGLE/V,N,REACT $
40. PURGE    KRR,KLR,QR,DM/REACT/GM/MPCF1/GO,KOO,L'OO,PO,UOOV,RUOV/OMIT/PS,
             KFS,KSS/SINGLE/QG/NOSR $
41. GPSP     GPL,GPST,USSET,SIL/OGPST/S,N,NOGPST $
42. COND     LBL4,NOGPST $
43. OFP      OGPST,,,,,// $

```

APPENDIX

44. LABEL LBL4 \$  
45. EQUIV KNN,KFF/SINGLE \$  
46. COND LBL3,SINGLE \$  
47. SCE1 USET,KNN,,,/KFF,KFS,KSS,,, \$  
48. LABEL LBL3 \$  
49. EQUIV KFF,AAA/OMIT \$  
50. COND LBL5,OMIT \$  
51. SMP1 USET,KFF,,,/GO,AAA,KOO,LOO,,,,, \$  
52. LABEL LBL5 \$  
53. EQUIV AAA,KLL/REACT \$  
54. COND LBL6,REACT \$  
55. RBMG1 USET,AAA,/KLL,KLR,KRR,,, \$  
56. LABEL LBL6 \$  
57. RBMG2 KLL/LLL \$  
58. COND LBL7,REACT \$  
59. RBMG3 LLL,KLR,KRR/DM \$  
60. LABEL LBL7 \$  
61. SSG1 SLT,BGPD,CSTM,SIL,EST,MPT, ,EDT,MGG,CASECC,DIT/PG/V,N,  
LUSET/V,N,NSKIP \$  
62. EQUIV PG,PL/NOSET \$  
63. COND LBL10,NOSET \$  
64. COND LBL10,NOSET \$  
65. SSG2 USET, ,YS,KFS,GO,DM,PG/QR,PO,PS,PL \$  
66. LABEL LBL10 \$  
67. SSG3 LLL,KLL,PL,LOO,KOO,PO/ULV,UOOV,RULV,RUOV/V,N,OMIT/V,Y,IRES=-1/  
V,N,NSKIP/S,N,EPSI \$  
68. SDR1 USET,PG,ULV,UOOV,YS,GO, ,PS,KFS,KSS,QR/UGV,PGG,QG/V,N,NSKIP/  
C,N,STATICS \$  
69. COND LBL8,REPEAT \$  
70. REPT LBL11,100  
71. JUMP ERROR1 \$  
72. PARAM //C,N,NOT/V,N,TEST/V,N,REPEAT \$  
73. COND ERROR5,TEST \$  
74. LABEL LBL8 \$  
75. SDR2 CASECC,CSTM,MPT,DIT,EQEXIN,SIL, ,EDT,BGPD,,QG,UGV,EST,  
,PGG/OPG1,OQG1,OUGV1,OES1,DEF1,PUGV1/C,N,STATICS/S,N,  
NOSORT2=-1 \$  
76. OFP OUGV1,OPG1,OQG1,DEF1,OES1,//S,N,CARDND \$  
77. COND P2,JUMPPLOT \$  
78. PLOT PLTPAR,GPSETS,ELSETS,CASECC,BGPD,EQEXIN,SIL,PUGV1,,GPECT,OES1/  
PLOTX2/V,N,NSIL/V,N,LUSET/V,N,JUMPPLOT/V,N,PLTFLG/S,N,PFILE \$  
79. PRTMSG PLOTX2// \$  
80. LABEL P2 \$  
81. JUMP FINIS \$  
82. LABEL ERROR1 \$  
83. PRTPARM //C,N,-1/C,N,STATICS \$  
84. LABEL ERROR3 \$  
85. PRTPARM //C,N,-3/C,N,STATICS \$  
86. LABEL ERROR4 \$  
87. PRTPARM //C,N,-4/C,N,STATICS \$  
88. LABEL ERROR5 \$  
89. PRTPARM //C,N,-5/C,N,STATICS \$  
90. LABEL FINIS \$  
91. END \$

## APPENDIX

### Description of DMAP Operations for Static Analysis

4. GP1 generates coordinate system transformation matrices, tables of grid-point locations, and tables for relating internal and external grid-point numbers.
5. GP2 generates an element connection table with internal indices.
8. Go to DMAP no. 16 if no plot package is present.
9. PLTSET transforms user input into a form used to drive structure plotter.
10. PRTMSG prints error messages associated with structure plotter.
13. Go to DMAP no. 16 if no undeformed structure plot request.
14. PLØT generates all requested undeformed structure plots.
15. PRTMSG prints plotter data and engineering data for each undeformed plot generated.
17. GP3 generates static loads table.
19. TAL generates element tables for use in matrix assembly and stress recovery.
21. Go to DMAP no. 85 and print error message if no elements have been defined.
24. Go to DMAP no. 33 if there are no structural elements.
27. EMG generates structural element matrix tables and dictionaries for later assembly.
28. Go to DMAP no. 30 if no stiffness matrix is to be assembled.
29. EMA assembles stiffness matrix  $[K_{gg}^x]$  and grid-point singularity table.
31. Go to DMAP no. 33 if no mass matrix is to be assembled.
32. EMA assembles mass matrix  $[M_{gg}]$ .
35. Go to next DMAP instruction if cold start.
36. Beginning of loop for additional constraint sets.
37. GP4 generates flags defining members of various displacement sets (USET), forms multipoint constraint equations  $[R_g]\{u_g\} = 0$ , and forms enforced displacement vector  $\{Y_s\}$ .
38. Go to DMAP no. 83 and print error message if no independent degrees of freedom are defined.

APPENDIX

41. GPSP determines if possible grid-point singularities remain.
42. Go to DMAP no. 44 if no grid-point singularities remain.
43. ØFP formats the table of possible grid-point singularities and places it on the system output file for printing.
45. Equivalence  $[K_{nn}]$  to  $[K_{ff}]$  if no single-point constraints.
46. Go to DMAP no. 48 if no single-point constraints.
47. SCE1 partitions out single-point constraints

$$[K_{nn}] = \left[ \begin{array}{c|c} K_{ff} & K_{fs} \\ \hline K_{sf} & K_{ss} \end{array} \right]$$

49. Equivalence  $[K_{ff}]$  to  $[K_{aa}]$  if no omitted coordinates.
50. Go to DMAP no. 52 if no omitted coordinates.
51. SMP1 partitions constrained stiffness matrix

$$[K_{ff}] = \left[ \begin{array}{c|c} \bar{K}_{aa} & K_{ao} \\ \hline K_{oa} & K_{oo} \end{array} \right]$$

solves for transformation matrix  $[G_o] = -[K_{oo}]^{-1}[K_{oa}]$ , and performs matrix reduction  $[K_{aa}] = [\bar{K}_{aa}] + [K_{oa}^T][G_o]$ .

53. Equivalence  $[K_{aa}]$  to  $[K_{ll}]$  if no free-body supports.
54. Go to DMAP no. 56 if no free-body supports.
55. RBMG1 partitions out free-body supports

$$[K_{aa}] = \left[ \begin{array}{c|c} K_{ll} & K_{lr} \\ \hline K_{rl} & K_{rr} \end{array} \right]$$

57. RBMG2 decomposes constrained stiffness matrix  $[K_{ll}] = [L_{ll}][U_{ll}]$ .
58. Go to DMAP no. 60 if no free-body supports.



APPENDIX

59. RBMG3 forms rigid-body transformation matrix

$$[D] = -[K_{\ell\ell}]^{-1}[K_{\ell r}]$$

calculates rigid-body check matrix

$$[X] = [K_{rr}] + [K_{\ell r}^T][D]$$

and calculates rigid-body error ratio

$$\epsilon = \frac{\|X\|}{\|K_{rr}\|}$$

61. SSG1 generates static load vectors  $\{P_g\}$ .

62. Equivalence  $\{P_g\}$  to  $\{P_\ell\}$  if no constraints applied.

64. SSG2 applies constraints to static load vectors

$$\begin{aligned} \{P_g\} &= \begin{Bmatrix} \bar{P}_n \\ - \\ \bar{P}_m \end{Bmatrix} & \{P_n\} &= \{\bar{P}_n\} + [G_m^T]\{P_m\} \\ \{P_n\} &= \begin{Bmatrix} \bar{P}_f \\ - \\ \bar{P}_s \end{Bmatrix} & \{P_f\} &= \{\bar{P}_f\} - [K_{fs}]\{Y_s\} \\ \{P_f\} &= \begin{Bmatrix} \bar{P}_a \\ - \\ \bar{P}_o \end{Bmatrix} & \{P_a\} &= \{\bar{P}_a\} + [G_o^T]\{P_o\} \\ \{P_a\} &= \begin{Bmatrix} P_\ell \\ - \\ P_r \end{Bmatrix} \end{aligned}$$

and calculates determinate forces of reaction  $\{Q_r\} = -\{P_r\} - [D^T]\{P_\ell\}$ .

66. SSG3 solves for displacements of independent coordinates

$$\{u_\ell\} = [K_{\ell\ell}]^{-1}\{P_\ell\}$$

APPENDIX

solves for displacements of omitted coordinates

$$\{u_o^o\} = [K_{oo}]^{-1}\{P_o\}$$

calculates residual vector (RULV) and residual vector error ratio for independent coordinates

$$\{\delta P_\ell\} = \{P_\ell\} - [K_{\ell\ell}]\{u_\ell\}$$

$$\epsilon_\ell = \frac{\begin{Bmatrix} u_\ell^T \end{Bmatrix} \{\delta P_\ell\}}{\begin{Bmatrix} P_\ell^T \end{Bmatrix} \{u_\ell\}}$$

and calculates residual vector (RUOV) and residual vector error ratio for omitted coordinates

$$\{\delta P_o\} = \{P_o\} - [K_{oo}]\{u_o^o\}$$

$$\epsilon_o = \frac{\begin{Bmatrix} u_o^o T \end{Bmatrix} \{\delta P_o\}}{\begin{Bmatrix} P_o \end{Bmatrix} \begin{Bmatrix} u_o^o \end{Bmatrix}}$$

67. SDR1 recovers dependent displacements

$$\begin{Bmatrix} u_\ell \\ - \\ u_r \end{Bmatrix} = \{u_a\} \qquad \{u_o\} = [G_o]\{u_a\} + \{u_o^o\}$$

$$\begin{Bmatrix} u_a \\ - \\ u_o \end{Bmatrix} = \{u_f\} \qquad \begin{Bmatrix} u_f \\ - \\ Y_s \end{Bmatrix} = \{u_n\}$$

$$\{u_m\} = [G_m]\{u_n\} \qquad \begin{Bmatrix} u_n \\ - \\ u_m \end{Bmatrix} = \{u_g\}$$

and recovers single-point forces of constraint

$$\{q_s\} = -\{P_s\} + [K_{fs}^T]\{u_f\} + [K_{ss}]\{Y_s\}$$

## APPENDIX

68. Go to DMAP no. 73 if all constraint sets have been processed.
69. Go to DMAP no. 36 if additional sets of constraints need to be processed.
70. Go to DMAP no. 81 and print error message if number of loops exceeds 100.
72. Go to DMAP no. 87 and print error message if multiple boundary conditions are attempted with improper subset.
74. SDR2 calculates element forces and stresses ( $\emptyset EF1$ ,  $\emptyset ES1$ ) and prepares load vectors, displacement vectors, and single-point forces of constraint for output ( $\emptyset PGL$ ,  $\emptyset UGV1$ ,  $PUGV1$ ,  $\emptyset \emptyset G1$ ).
75.  $\emptyset FP$  formats tables prepared by SDR2 and places them on the system output file for printing.
76. Go to DMAP no. 79 if no deformed structure plots are requested.
77.  $PL\emptyset T$  generates all requested deformed structure plots.
79.  $PRMSG$  prints plotter data and engineering data for each deformed plot generated.
80. Go to DMAP no. 89 and make a normal exit.
82. STATIC ANALYSIS ERROR MESSAGE NO. 1 - ATTEMPT TO EXECUTE MORE THAN 100 LOOPS.
84. STATIC ANALYSIS ERROR MESSAGE NO. 3 - NO INDEPENDENT DEGREES OF FREEDOM HAVE BEEN DEFINED.
86. STATIC ANALYSIS ERROR MESSAGE NO. 4 - NO ELEMENTS HAVE BEEN DEFINED.
88. STATIC ANALYSIS ERROR MESSAGE NO. 5 - A LOOPING PROBLEM RUN ON NON-LOOPING SUBSET.

### Static Analysis Sample Problem

JANUARY 21, 1977 NASTRAN 1/ 1/77 PAGE 1

N A S T R A N   E X E C U T I V E   C O N T R O L   D E C K   E C H O

```
ID      MINITRAN.DEMO-1
APP     DMAP
XDMAP   ERR=2,LIST,REF $  STATICS ANALYSIS
(SEE NASTRAN SOURCE PROGRAM COMPILATION FOR LISTING OF DMAP SEQUENCE)
END     $
TIME    3
DIAG 1
CEND
```

# APPENDIX

DEMONSTRATION PROBLEM - STATIC ANALYSIS  
DEFLECTION OF A UNIFORM BEAM

JANUARY 21, 1977 NASTRAN 1/ 1/77 PAGE 2

```

CASE CONTROL DECK ECHO
CARD
COUNT
1 TITLE = DEMONSTRATION PROBLEM - STATIC ANALYSIS
2 SUBTITLE = DEFLECTION OF AN UNIFORM BEAM
3 ECHO = BOTH
4 LINE = 40
5 OUTPUT
6 SET 3 = 1,6,7,8,10
7 DISP = ALL
8 OLOAD = ALL
9 SPCF = ALL
10 STRESS = 3
11 SUBCASE 1
12 LABEL = CANTILEVER BEAM SUBJECTED TO CONCENTRATED FORCE
13 SPC = 10
14 LOAD = 10
15 SUBCASE 2
16 LABEL = CANTILEVER BEAM UNDER MOMENT AT FREE END
17 SPC = 10
18 LOAD = 12
19 SUBCASE 3
20 LABEL = SIMPLY-SUPPORTED BEAM WITH CONCENTRATED FORCE
21 SPC = 1
22 LOAD = 20
23 BEGIN BULK
    
```

DEMONSTRATION PROBLEM - STATIC ANALYSIS  
DEFLECTION OF A UNIFORM BEAM

JANUARY 21, 1977 NASTRAN 1/ 1/77 PAGE 3

```

INPUT BULK DATA DECK ECHO
. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9 .. 10 .
$ STRUCTURAL ELEMENTS
CBAR 1 39 1 2 0.0 1.0 0.0 1
CBAR 2 39 2 3 0.0 1.0 0.0 1
CBAR 3 39 3 4 0.0 1.0 0.0 1
CBAR 4 39 4 5 0.0 1.0 0.0 1
CBAR 5 39 5 6 0.0 1.0 0.0 1
CBAR 6 39 6 7 0.0 1.0 0.0 1
CBAR 7 39 7 8 0.0 1.0 0.0 1
CBAR 8 39 8 9 0.0 1.0 0.0 1
CBAR 9 39 9 10 0.0 1.0 0.0 1
CBAR 10 39 10 11 0.0 1.0 0.0 1
CNGRNT 1 2 THRU 10
PBAR 39 72 18.0 54.0 +B
+8 3.0 0.0
MAT1 72 1.0+6
$ GEOMETRY
GRID 1 0.0 1345
GRID 2 24.0 1345
GRID 3 48.0 1345
GRID 4 72.0 1345
GRID 5 96.0 1345
GRID 6 120.0 1345
GRID 7 144.0 1345
GRID 8 168.0 1345
GRID 9 192.0 1345
GRID 10 216.0 1345
GRID 11 240.0 1345
$ CONSTRAINT AND LOADING FOR CASE 1
SPC 10 1 26
FORCE 10 9 30.0 0.0 -1.0 0.0
$ LOADING FOR CASE 2
MOMENT 12 11 2100.0 0.0 0.0 -1.0
$ CONSTRAINT AND LOADING FOR CASE 3
SPC 20 1 2
SPC 21 11 2
SPCADD 1 20 21
FORCE 20 7 -240.0 0.0 1.0 0.0
ENDDATA
TOTAL COUNT= 38
    
```

# APPENDIX

\*\*NO ERRORS FOUND - EXECUTE NASTRAN PROGRAM\*\*

\*\*\* SYSTEM INFORMATION MESSAGE 3113, EMGPRO PROCESSING SINGLE PRECISION ELEMENTS OF TYPE 34 STARTING WITH ID 1

\*\*\* SYSTEM INFORMATION MESSAGE 3107, EMGOLD IS PROCESSING ELEMENTS OF TYPE = 34, BEGINNING WITH ELEMENT ID = 1

\*\*\*USER INFORMATION MESSAGE 3023--PARAMETERS FOR SYMMETRIC DECOMPOSITION OF DATA BLOCK KLL ( N = 20 )  
 TIME ESTIMATE= 1 C AVG = 3 PC AVG = 0 SPILL GROUPS = 0 S AVG = 1  
 ADDITIONAL CORE= -16406 C MAX = 4 PCHAX = 0 PC GROUPS = 0 PREFACE LOOPS = 1  
 MPYAD--NULL MATRIX PRODUCT  
 METHOD 1 NT,NBR PASSES = 1,EST. TIME = 00.0

\*\*\* USER INFORMATION MESSAGE 3035

FOR LOAD 1 EPSILON SUB E = -2.5175003E-11

\*\*\* USER INFORMATION MESSAGE 3035

FOR LOAD 2 EPSILON SUB E = -1.5929497E-11

MPYAD--NULL MATRIX PRODUCT  
 METHOD 2 T ,NBR PASSES = 1,EST. TIME = 00.0

\*\*\*USER INFORMATION MESSAGE 3023--PARAMETERS FOR SYMMETRIC DECOMPOSITION OF DATA BLOCK KLL ( N = 20 )  
 TIME ESTIMATE= 1 C AVG = 3 PC AVG = 0 SPILL GROUPS = 0 S AVG = 1  
 ADDITIONAL CORE= -16406 C MAX = 4 PCHAX = 0 PC GROUPS = 0 PREFACE LOOPS = 1  
 MPYAD--NULL MATRIX PRODUCT  
 METHOD 1 NT,NBR PASSES = 1,EST. TIME = 00.0

\*\*\* USER INFORMATION MESSAGE 3035

FOR LOAD 3 EPSILON SUB E = -7.9936584E-12

MPYAD--NULL MATRIX PRODUCT  
 METHOD 2 T ,NBR PASSES = 1,EST. TIME = 00.0

\*\*\* SYSTEM WARNING MESSAGE 3022

DATA BLOCK PLTPAR IS REQUIRED AS INPUT AND IS NOT OUTPUT BY A PREVIOUS MODULE IN THE CURRENT DMAP ROUTE.

\*\*\* SYSTEM WARNING MESSAGE 3022

DATA BLOCK GPSETS IS REQUIRED AS INPUT AND IS NOT OUTPUT BY A PREVIOUS MODULE IN THE CURRENT DMAP ROUTE.

\*\*\* SYSTEM WARNING MESSAGE 3022

DATA BLOCK ELSETS IS REQUIRED AS INPUT AND IS NOT OUTPUT BY A PREVIOUS MODULE IN THE CURRENT DMAP ROUTE.

DEMONSTRATION PROBLEM - STATIC ANALYSIS JANUARY 21, 1977 NASTRAN 1/ 1/77 PAGE 11  
 DEFLECTION OF A UNIFORM BEAM

CANTILEVER BEAM SUBJECTED TO CONCENTRATED FORCE SUBCASE 1

D I S P L A C E M E N T V E C T O R										
POINT ID.	TYPE	T1	T2	T3	R1	R2	R3			
1	G	0.0	0.0	0.0	0.0	0.0	0.0			
2	G	0.0	-2.944000E-02	0.0	0.0	0.0	-2.400000E-03			
3	G	0.0	-1.126400E-01	0.0	0.0	0.0	-4.480000E-03			
4	G	0.0	-2.419200E-01	0.0	0.0	0.0	-6.240000E-03			
5	G	0.0	-4.096000E-01	0.0	0.0	0.0	-7.680000E-03			
6	G	0.0	-6.080000E-01	0.0	0.0	0.0	-8.800000E-03			
7	G	0.0	-8.294400E-01	0.0	0.0	0.0	-9.600000E-03			
8	G	0.0	-1.066240E+00	0.0	0.0	0.0	-1.008000E-02			
9	G	0.0	-1.310720E+00	0.0	0.0	0.0	-1.024000E-02			
10	G	0.0	-1.556480E+00	0.0	0.0	0.0	-1.024000E-02			
11	G	0.0	-1.802240E+00	0.0	0.0	0.0	-1.024000E-02			

DEMONSTRATION PROBLEM - STATIC ANALYSIS JANUARY 21, 1977 NASTRAN 1/ 1/77 PAGE 12  
 DEFLECTION OF A UNIFORM BEAM

CANTILEVER BEAM UNDER MOMENT AT FREE END SUBCASE 2

D I S P L A C E M E N T V E C T O R										
POINT ID.	TYPE	T1	T2	T3	R1	R2	R3			
1	G	0.0	0.0	0.0	0.0	0.0	0.0			
2	G	0.0	-1.120000E-02	0.0	0.0	0.0	-9.333333E-04			
3	G	0.0	-4.480000E-02	0.0	0.0	0.0	-1.866667E-03			
4	G	0.0	-1.008000E-01	0.0	0.0	0.0	-2.800000E-03			
5	G	0.0	-1.792000E-01	0.0	0.0	0.0	-3.733333E-03			
6	G	0.0	-2.800000E-01	0.0	0.0	0.0	-4.666667E-03			
7	G	0.0	-4.032000E-01	0.0	0.0	0.0	-5.600000E-03			
8	G	0.0	-5.488000E-01	0.0	0.0	0.0	-6.533333E-03			
9	G	0.0	-7.168000E-01	0.0	0.0	0.0	-7.466667E-03			
10	G	0.0	-9.072000E-01	0.0	0.0	0.0	-8.400000E-03			
11	G	0.0	-1.120000E+00	0.0	0.0	0.0	-9.333333E-03			



# APPENDIX

DEMONSTRATION PROBLEM - STATIC ANALYSIS  
DEFLECTION OF A UNIFORM BEAM

JANUARY 21, 1977 NASTRAN 1/ 1/77 PAGE 19

SIMPLY-SUPPORTED BEAM WITH CONCENTRATED FORCE

SUBCASE 3

FORCES OF SINGLE-POINT CONSTRAINT

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
1	G	0.0	9.600000E+01	0.0	0.0	0.0	0.0
11	G	0.0	1.440000E+02	0.0	0.0	0.0	0.0

DEMONSTRATION PROBLEM - STATIC ANALYSIS  
DEFLECTION OF A UNIFORM BEAM

JANUARY 21, 1977 NASTRAN 1/ 1/77 PAGE 20

CANTILEVER BEAM SUBJECTED TO CONCENTRATED FORCE

SUBCASE 1

ELEMENT ID.	SA1 SB1	STRESSES IN BAR ELEMENTS			AXIAL STRESS	(C B A R)		M.S.-T M.S.-C
		SA2 SB2	SA3 SB3	SA4 SB4		SA-MAX SB-MAX	SA-MIN SB-MIN	
1	3.200000E+02	0.0	0.0	0.0	0.0	3.200000E+02	0.0	
	2.800000E+02	0.0	0.0	0.0	0.0	2.800000E+02	0.0	
6	1.200000E+02	0.0	0.0	0.0	0.0	1.200000E+02	0.0	
	8.000000E+01	0.0	0.0	0.0	0.0	8.000000E+01	0.0	
7	8.000000E+01	0.0	0.0	0.0	0.0	8.000000E+01	0.0	
	4.000000E+01	0.0	0.0	0.0	0.0	4.000000E+01	0.0	
8	4.000000E+01	0.0	0.0	0.0	0.0	4.000000E+01	0.0	
	-2.069606E-10	0.0	0.0	0.0	0.0	0.0	-2.069606E-10	
10	2.069606E-10	0.0	0.0	0.0	0.0	2.069606E-10	0.0	
	-4.139211E-10	0.0	0.0	0.0	0.0	0.0	-4.139211E-10	

DEMONSTRATION PROBLEM - STATIC ANALYSIS  
DEFLECTION OF A UNIFORM BEAM

JANUARY 21, 1977 NASTRAN 1/ 1/77 PAGE 21

CANTILEVER BEAM UNDER MOMENT AT FREE END

SUBCASE 2

ELEMENT ID.	SA1 SB1	STRESSES IN BAR ELEMENTS			AXIAL STRESS	(C B A R)		M.S.-T M.S.-C
		SA2 SB2	SA3 SB3	SA4 SB4		SA-MAX SB-MAX	SA-MIN SB-MIN	
1	1.166667E+02	0.0	0.0	0.0	0.0	1.166667E+02	0.0	
	1.166667E+02	0.0	0.0	0.0	0.0	1.166667E+02	0.0	
6	1.166667E+02	0.0	0.0	0.0	0.0	1.166667E+02	0.0	
	1.166667E+02	0.0	0.0	0.0	0.0	1.166667E+02	0.0	
7	1.166667E+02	0.0	0.0	0.0	0.0	1.166667E+02	0.0	
	1.166667E+02	0.0	0.0	0.0	0.0	1.166667E+02	0.0	
8	1.166667E+02	0.0	0.0	0.0	0.0	1.166667E+02	0.0	
	1.166667E+02	0.0	0.0	0.0	0.0	1.166667E+02	0.0	
10	1.166667E+02	0.0	0.0	0.0	0.0	1.166667E+02	0.0	
	1.166667E+02	0.0	0.0	0.0	0.0	1.166667E+02	0.0	

DEMONSTRATION PROBLEM - STATIC ANALYSIS  
DEFLECTION OF A UNIFORM BEAM

JANUARY 21, 1977 NASTRAN 1/ 1/77 PAGE 22

SIMPLY-SUPPORTED BEAM WITH CONCENTRATED FORCE

SUBCASE 3

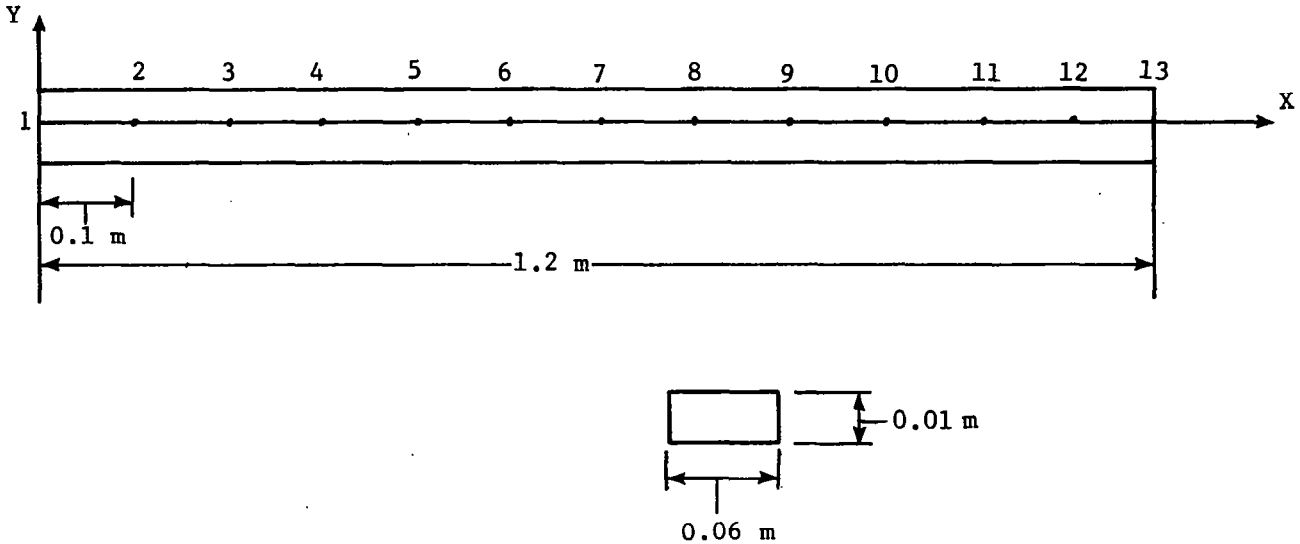
ELEMENT ID.	SA1 SB1	STRESSES IN BAR ELEMENTS			AXIAL STRESS	(C B A R)		M.S.-T M.S.-C
		SA2 SB2	SA3 SB3	SA4 SB4		SA-MAX SB-MAX	SA-MIN SB-MIN	
1	2.587007E-11	0.0	0.0	0.0	0.0	2.587007E-11	0.0	
	-1.280000E+02	0.0	0.0	0.0	0.0	0.0	-1.280000E+02	
6	-6.400000E+02	0.0	0.0	0.0	0.0	0.0	-6.400000E+02	
	-7.680000E+02	0.0	0.0	0.0	0.0	0.0	-7.680000E+02	
7	-7.680000E+02	0.0	0.0	0.0	0.0	0.0	-7.680000E+02	
	-5.760000E+02	0.0	0.0	0.0	0.0	0.0	-5.760000E+02	
8	-5.760000E+02	0.0	0.0	0.0	0.0	0.0	-5.760000E+02	
	-3.840000E+02	0.0	0.0	0.0	0.0	0.0	-3.840000E+02	
10	-1.920000E+02	0.0	0.0	0.0	0.0	0.0	-1.920000E+02	
	-5.174014E-11	0.0	0.0	0.0	0.0	0.0	-5.174014E-11	

## APPENDIX

### Vibration Analysis

This problem is a vibration analysis of a cantilever beam.

Model



### Vibration Analysis DMAP Sequence

```

1. XDMAP      ERR=2,LIST,REF $  VIBRATION ANALYSIS
2. FILE      LAMA=APPEND/PHIA=APPEND $
3. GP1       GEOM1,GEOM2,/GPL,EQEXIN,GPDT,CSTM,BGPDT,SIL/S,N,LUSET/  V,N,
              NOGPDT $
4. GP2       GEOM2,EQEXIN/ECT $
5. PARAML    PCDB//*PRES*///NOPCDB $
6. PURGE     PLTSETX,PLTPAR,GPSETS,ELSETS/NOPCDB $
7. COND      P1,NOPCDB $
8. PLTSET    PCDB,EQEXIN,ECT/PLTSETX,PLTPAR,GPSETS,ELSETS/S,N,NSIL/S,N,
              JUMPPLOT=-1 $
9. PRTMSG    PLTSETX// $
10. PARAM    //*MPY*/PLTFLG/1/1 $
11. PARAM    //*MPY*/PFILE/0/0 $
12. COND      P1,JUMPPLOT $
13. PLOT      PLTPAR,GPSETS,ELSETS,CASECC,BGPDT,EQEXIN,SIL/
              PLOTX1/NSIL/LUSET/S,N,JUMPPLOT/S,N,PLTFLG/S,N,PFILE $
14. PRTMSG    PLOTX1// $
15. LABEL    P1 $
16. TAIL     ECT,EPT,BGPDT,SIL,      ,CSTM/EST,GEI,GPECT,/V,N,LUSET/  S,N,
              NOSIMP/C,N,1/S,N,NOGENL/S,N,GENEL $
17. COND      ERROR1,NOSIMP $
18. PURGE     OGPST/GENEL $
19. PARAM    //C,N,ADD/V,N,NOKGGX/C,N,1/C,N,0 $
20. PARAM    //C,N,ADD/V,N,NOMGG/C,N,1/C,N,0 $
21. EMG      EST,CSTM,MPT,DIT,GEOM2,/KELM,KDICT,MELM,MDICT,,/S,N,NOKGGX/ S,
              N,NOMGG/C,N,/C,N,/C,N,/C,Y,COUPMASS/C,Y,CPBAR/C,Y,CPROD/C,Y,
              CPQUAD1/C,Y,CPQUAD2/C,Y,CPTRIA1/C,Y,CPTRIA2/  C,Y,CPTUBE/C,Y,
              CPQDPLT/C,Y,CPTRPLT/C,Y,CPTRBSC $

```



APPENDIX

22. COND JMPKGG,NOKGGX \$  
23. EMA GPECT,KDICT,KELM/KNN,GPST \$  
24. LABEL JMPKGG \$  
25. COND ERROR1,NOMGG \$  
26. EMA GPECT,MDICT,MELM/MNN,/C,N,-1/C,Y,WTMASS=1.0 \$  
27. PARAM //C,N,MPY/V,N,NSKIP/C,N,0/C,N,0 \$  
28. GP4 CASECC,GEOM4,EQEXIN,SIL,GPDT,BGPDT,CSTM/RG,,USET,ASET/ V,N,  
LUSET/S,N,MPCF1/S,N,MPCF2/S,N,SINGLE/S,N,OMIT/S,N,REACT/S,N,  
NSKIP/S,N,REPEAT/S,N,NOSET/S,N,NOL/S,N,NOA/C,Y,SUBID \$  
29. COND ERROR3,NOL \$  
30. PURGE KRR,KLR,DM,MLR,MR/REACT/GM/MPCF1/GO/OMIT/KFS/SINGLE/QG/NOSET \$  
31. GPSP GPL,GPST,USET,SIL/OGPST/S,N,NOGPST \$  
32. COND LBL4,NOGPST \$  
33. OFP OGPST,,,,,// \$  
34. LABEL LBL4 \$  
35. EQUIV KNN,KFF/SINGLE/MNN,MFF/SINGLE \$  
36. COND LBL3,SINGLE \$  
37. SCE1 USET,KNN,MNN,,/KFF,KFS,,MFF,, \$  
38. LABEL LBL3 \$  
39. EQUIV KFF,KAA/OMIT \$  
40. EQUIV MFF,MAA/OMIT \$  
41. COND LBL5,OMIT \$  
42. SMP1 USET,KFF,,,/GO,KAA,KOO,LOO,,,, \$  
43. SMP2 USET,GO,MFF/MAA \$  
44. LABEL LBL5 \$  
45. COND LBL6,REACT \$  
46. RBMG1 USET,KAA,MAA/KLL,KLR,KRR,MLL,MLR,MRR \$  
47. RBMG2 KLL/LLL \$  
48. RBMG3 LLL,KLR,KRR/DM \$  
49. RBMG4 DM,MLL,MLR,MRR/MR \$  
50. LABEL LBL6 \$  
51. DPD DYNAMICS,GPL,SIL,USET/GPLD,SILD,USETD,,,,,EED,EQDYN/V,N,  
LUSET/V,N,LUSETD/V,N,NOTFL/V,N,NODLT/V,N,NOPSDL/V,N,NOFRL/ V,  
N,NONLFT/V,N,NOTRL/S,N,NOEED/C,N,/V,N,NOUE \$  
52. COND ERROR2,NOEED \$  
53. PARAM //C,N,MPY/V,N,NEIGV/C,N,1/C,N,-1 \$  
54. READ KAA,MAA,MR,DM,EED,USET,CASECC/LAMA,PHIA,MI,OEIGS/C,N,MODES/S,N,  
NEIGV \$  
55. PARAM //C,N,MPY/V,N,CARDNO/C,N,0/C,N,0 \$  
56. OFP LAMA,OEIGS,,,,//S,N,CARDNO \$  
57. COND FINIS,NEIGV \$  
58. SDR1 USET,,PHIA,,,GO, , ,KFS,,/PHIG,,QG/C,N,1/C,N,REIG \$  
59. PARAM /\*MPY\*/SIXSIL/NSIL/6 \$  
60. PARAM /\*EQ\*/SCALAR/SIXSIL/LUSET \$  
61. EQUIV SIL,SIP/SCALAR/BGPDT,BGPDP/SCALAR \$  
62. SDR2 CASECC,CSTM,MPT,DIT,EQEXIN,SIL,,,BGPDP,LAMA,QG,PHIG,EST,,/ ,  
OQG1,OPHIG,OES1,DEF1,PPHIG/C,N,REIG \$  
63. OFP OPHIG,OQG1,DEF1,OES1,,,/S,N,CARDNO \$  
64. COND P2,JUMPPLOT \$  
65. PLQT PLTPAR,GPSETS,ELSETS,CASECC,BGPDP,EQEXIN,SIP,,PPHIG,GPECT,  
OES1/PLOTX2/NSIL/LUSET/JUMPPLOT/PLTFLG/S,N,PFILE \$  
66. PRMSG PLOTX2// \$  
67. LABEL P2 \$  
68. JUMP FINIS \$

## APPENDIX

```
69. LABEL      ERROR1 $
70. PRTPARM    //C,N,-1/C,N,MODES $
71. LABEL      ERROR2 $
72. PRTPARM    //C,N,-2/C,N,MODES $
73. LABEL      ERROR3 $
74. PRTPARM    //C,N,-3/C,N,MODES $
75. LABEL      FINIS $
76. END        $
```

### Description of DMAP Operations for Vibration Analysis

3. GP1 generates coordinate system transformation matrices, tables of grid-point locations, and tables for relating internal and external grid-point numbers.
4. GP2 generates an element connection table with internal indices.
7. Go to DMAP no. 15 if no plot package is present.
8. PLTSET transforms user input into a form used to drive structure plotter.
9. PRTMSG prints error messages associated with structure plotter.
12. Go to DMAP no. 15 if no undeformed structure plot request.
13. PLØT generates all requested undeformed structure plots.
14. PRTMSG prints plotter data and engineering data for each undeformed plot generated.
16. TAL generates element tables for use in matrix assembly and stress recovery.
17. Go to DMAP no. 65 and print error message if there are no structural elements.
21. EMG generates structural element matrix tables and dictionaries for later assembly.
22. Go to DMAP no. 24 if no stiffness matrix is to be assembled.
23. EMA assembles stiffness matrix  $[K_{gg}^x]$  and grid-point singularity table.
25. Go to DMAP no. 69 and print error message if no mass matrix exists.
26. EMA assembles mass matrix  $[M_{gg}]$ .
28. GP4 generates flags defining numbers of various displacement sets (USET).
29. Go to DMAP no. 73 and print error message if no independent degrees of freedom are defined.

APPENDIX

31. GPSP determines if possible grid-point singularities remain.
32. Go to DMAP no. 34 if no grid-point singularity table.
33. ØFP formats table of possible grid-point singularities and places it on the system output file for printing.
35. Equivalence  $[K_{nn}]$  to  $[K_{ff}]$  and  $[M_{nn}]$  to  $[M_{ff}]$  if no single-point constraints.
36. Go to DMAP no. 38 if no single-point constraints.
37. SCE1 partitions out single-point constraints

$$[K_{nn}] = \left[ \begin{array}{c|c} K_{ff} & K_{fs} \\ \hline K_{sf} & K_{ss} \end{array} \right] \quad \text{and} \quad [M_{nn}] = \left[ \begin{array}{c|c} M_{ff} & M_{fs} \\ \hline M_{sf} & M_{ss} \end{array} \right]$$

39. Equivalence  $[K_{ff}]$  to  $[K_{aa}]$  if no omitted coordinates.
40. Equivalence  $[M_{ff}]$  to  $[M_{aa}]$  if no omitted coordinates.
41. Go to DMAP no. 49 if no omitted coordinates.
42. SMP1 partitions constrained stiffness matrix

$$[K_{ff}] = \left[ \begin{array}{c|c} \bar{K}_{aa} & K_{ao} \\ \hline K_{oa} & K_{oo} \end{array} \right]$$

solves for transformation matrix  $[G_o] = -[K_{oo}]^{-1}[K_{oa}]$ , and performs matrix reduction  $[K_{aa}] = [\bar{K}_{aa}] + [K_{oa}^T][G_o]$ .

43. SMP2 partitions constrained mass matrix

$$[M_{ff}] = \left[ \begin{array}{c|c} \bar{M}_{aa} & M_{ao} \\ \hline M_{oa} & M_{oo} \end{array} \right]$$

and performs matrix reduction

$$[M_{aa}] = [\bar{M}_{aa}] + [M_{oa}^T][G_o] + [G_o^T][M_{oa}] + [G_o^T][M_{oo}][G_o]$$

APPENDIX

45. Go to DMAP no. 50 if no free-body supports.

46. RBMG1 partitions out free-body supports

$$[K_{aa}] = \begin{bmatrix} K_{\ell\ell} & | & K_{\ell r} \\ \hline K_{r\ell} & | & K_{rr} \end{bmatrix} \quad \text{and} \quad [M_{aa}] = \begin{bmatrix} M_{\ell\ell} & | & M_{\ell r} \\ \hline M_{r\ell} & | & M_{rr} \end{bmatrix}$$

47. RBMG2 decomposes constrained stiffness matrix  $[K_{\ell\ell}] = [L_{\ell\ell}][U_{\ell\ell}]$ .

48. RBMG3 forms rigid-body transformation matrix

$$[D] = -[K_{\ell\ell}]^{-1}[K_{\ell r}]$$

calculates rigid-body check matrix

$$[X] = [K_{rr}] + [K_{\ell r}^T][D]$$

and calculates rigid-body error ratio

$$\epsilon = \frac{\|X\|}{\|K_{rr}\|}$$

49. RBMG4 forms rigid-body mass matrix

$$[m_r] = [M_{rr}] + [M_{\ell r}^T][D] + [D^T][M_{\ell r}] + [D^T][M_{\ell\ell}][D]$$

51. DPD extracts eigenvalue extraction data from dynamics data block.

52. Go to DMAP no. 71 and print error message if no eigenvalue extraction data.

54. READ extracts real eigenvalues from the equation

$$[K_{aa} - \lambda M_{aa}] \{u_a\} = 0$$

APPENDIX

calculates rigid-body modes by finding a square matrix  $[\phi_{ro}]$  such that

$$[m_o] = [\phi_{ro}^T] [m_r] [\phi_{ro}]$$

is diagonal and normalized, computes rigid-body eigenvectors

$$[\phi_{ao}] = \begin{bmatrix} D & \phi_{ro} \\ - & - \\ \phi_{ro} & \end{bmatrix}$$

calculates modal mass matrix

$$[m] = [\phi_a^T] [M_{aa}] [\phi_a]$$

and normalizes eigenvectors according to one of the following user requests:

Unit value of selected coordinate

Unit value of largest component

Unit value of generalized mass

56. ØFP formats eigenvalues and summary of eigenvalue extraction information and places them on the system output file for printing.

57. Go to DMAP no. 75 and exit if no eigenvalues found.

58. SDR1 recovers dependent components of the eigenvectors

$$\{\phi_o\} = [G_o] \{\phi_a\} \quad \begin{Bmatrix} \phi_a \\ - \\ \phi_o \end{Bmatrix} = \{\phi_f\}$$

$$\begin{Bmatrix} \phi_f \\ - \\ \phi_s \end{Bmatrix} = \{\phi_n\} \quad \{\phi_m\} = [G_m] \{\phi_n\}$$

$$\begin{Bmatrix} \phi_n \\ - \\ \phi_m \end{Bmatrix} = \{\phi_g\}$$

and recovers single-point forces of constraint  $\{q_s\} = [K_{fs}]^T \{\phi_f\}$ .

APPENDIX

61. Equivalence SIL to SIP and BGPDT to BGPDP when one or more geometric grid points exist.
62. SDR2 calculates element forces and stresses ( $\emptyset EF1$ ,  $\emptyset ES1$ ) and prepares eigenvectors and single-point forces of constraint for output ( $\emptyset PHIG$ ,  $PPHIG$ ,  $\emptyset QG1$ ).
63.  $\emptyset FP$  formats tables prepared by SDR2 and places them on the system output file for printing.
64. Go to DMAP no. 67 if no deformed structure plots are requested.
65.  $PL\emptyset T$  generates all requested deformed structure plots.
66.  $PR\emptyset MSG$  prints plotter data and engineering data for each deformed plot generated.
68. Go to DMAP no. 75 and make normal exit.
70. ~~NORMAL MODE ANALYSIS ERROR MESSAGE NO. 1 - MASS MATRIX REQUIRED FOR REAL EIGENVALUE ANALYSIS.~~
72. ~~NORMAL MODE ANALYSIS ERROR MESSAGE NO. 2 - EIGENVALUE EXTRACTION DATA REQUIRED FOR REAL EIGENVALUE ANALYSIS.~~
74. ~~NORMAL MODE ANALYSIS ERROR MESSAGE NO. 3 - NO INDEPENDENT DEGREES OF FREEDOM HAVE BEEN DEFINED.~~

Vibration Analysis Sample Problem

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 1

N A S T R A N   F X E C U T I V E   C O N T R O L   D E C K   E C H O

```

ID      MINITRAN, DEMO-2R
APP     DMAP
XDMAP   ERR=2, LIST, PEF %  VIBRATION ANALYSIS
(SEE NASTRAN SOURCE PROGRAM COMPILATION FOR LISTING OF DMAP SEQUENCE)
END     $
TIME    4
CEND

```

DEMONSTRATION PROBLEM - VIBRATION ANALYSIS  
CANTILEVER BEAM - SECOND OF THREE PROBLEMS

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 2

C A S E   C O N T R O L   D E C K   E C H O

```

CARD
COUNT
1      TITLE = DEMONSTRATION PROBLEM - VIBRATION ANALYSIS
2      SURTITLE = CANTILEVER BEAM - SECOND OF THREE PROBLEMS
3      LINE = 40
4      ECHO = BOTH
5      METHOD = 12
6      SPC = 20
7      OUTPUT
8      DISP = ALL
9      BEGIN BULK

```

# APPENDIX

DEMONSTRATION PROBLEM - VIBRATION ANALYSIS  
CANTILEVER BEAM - SECOND OF THREE PROBLEMS

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 3

```

INPUT BULK DATA DECK ECHO
. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9 .. 10 .
% STRUCTURAL ELEMENTS
CBAR 11 201 1 2 0.0 1.0 0.0 1
CBAR 12 201 2 3 0.0 1.0 0.0 1
CBAR 13 201 3 4 0.0 1.0 0.0 1
CBAR 14 201 4 5 0.0 1.0 0.0 1
CBAR 15 201 5 6 0.0 1.0 0.0 1
CBAR 16 201 6 7 0.0 1.0 0.0 1
CBAR 17 201 7 8 0.0 1.0 0.0 1
CBAR 18 201 8 9 0.0 1.0 0.0 1
CBAR 19 201 9 10 0.0 1.0 0.0 1
CBAR 20 201 10 11 0.0 1.0 0.0 1
CBAR 21 201 11 12 0.0 1.0 0.0 1
CBAR 22 201 12 13 0.0 1.0 0.0 1
CNGPNT 11 12 THRU 22
PARAM COUPHASS1
PRAR 201 6 6.0-4 5.0-9
MAT1 6 7.0+10 0.3 86.0
% GEOMETRY
GRID 1 0.0 1345
GRID 2 0.1 1345
GRID 3 0.2 1345
GRID 4 0.3 1345
GRID 5 0.4 1345
GRID 6 0.5 1345
GRID 7 0.6 1345
GRID 8 0.7 1345
GRID 9 0.8 1345
GRID 10 0.9 1345
GRID 11 1.0 1345
GRID 12 1.1 1345
GRID 13 1.2 1345
% SUPPORT CONDITION
SPC 20 1 26
% EIGENVALUE METHOD
EIGR 12 INV 0.0 500.0 3 3 +8C
+BC MAX
% REDUCTION OF 0.0.F.
OMIT 13 6 2 6 3 6 4 6
OMIT 5 6 6 6 7 6 8 6
OMIT 9 6 10 6 11 6 12 6
LH>10
    
```

\*\*NO ERRORS FOUND - EXECUTE NASTRAN PROGRAM\*\*

```

*** SYSTEM INFORMATION MESSAGE 3113, EMGPRO PROCESSING SINGLE PRECISION ELEMENTS OF TYPE 34 STARTING WITH ID 11
*** SYSTEM INFORMATION MESSAGE 3107, EMGOLD IS PROCESSING ELEMENTS OF TYPE = 34, BEGINNING WITH ELEMENT ID = 11
***USER INFORMATION MESSAGE 3023--PARAMETERS FOR SYMMETRIC DECOMPOSITION OF DATA BLOCK K00 ( N = 12 )
TIME ESTIMATE= 1 C AVG = 1 PC AVG = 0 SPILL GROUPS = 0 S AVG = 1
ADDITIONAL CORE= -16415 C MAX = 2 PCMAX = 0 PC GROUPS = 0 PREFACE LOOPS = 1
METHOD 3 T ,NBR PASSES = 1,EST. TIME = 00.0
METHOD 1 NT,NBR PASSES = 1,EST. TIME = 00.0
METHOD 3 T ,NBR PASSES = 1,EST. TIME = 00.0
METHOD 3 T ,NBR PASSES = 1,EST. TIME = 00.0

***USER INFORMATION MESSAGE 3023--PARAMETERS FOR SYMMETRIC DECOMPOSITION OF DATA BLOCK LAMA ( N = 12 )
TIME ESTIMATE= 1 C AVG = 6 PC AVG = 0 SPILL GROUPS = 0 S AVG = 1
ADDITIONAL CORE= -14107 C MAX = 12 PCMAX = 0 PC GROUPS = 0 PREFACE LOOPS = 1

***USER INFORMATION MESSAGE 3023--PARAMETERS FOR SYMMETRIC DECOMPOSITION OF DATA BLOCK LAMA ( N = 12 )
TIME ESTIMATE= 1 C AVG = 6 PC AVG = 0 SPILL GROUPS = 0 S AVG = 1
ADDITIONAL CORE= -14107 C MAX = 12 PCMAX = 0 PC GROUPS = 0 PREFACE LOOPS = 1

***USER INFORMATION MESSAGE 3023--PARAMETERS FOR SYMMETRIC DECOMPOSITION OF DATA BLOCK LAMA ( N = 12 )
TIME ESTIMATE= 1 C AVG = 6 PC AVG = 0 SPILL GROUPS = 0 S AVG = 1
ADDITIONAL CORE= -14107 C MAX = 12 PCMAX = 0 PC GROUPS = 0 PREFACE LOOPS = 1
METHOD 1 NT,NBR PASSES = 1,EST. TIME = 00.0
METHOD 3 T ,NBR PASSES = 1,EST. TIME = 00.0
    
```

## APPENDIX

DEMONSTRATION PROBLEM - VIBRATION ANALYSIS  
CANTILEVER BEAM - SECOND OF THREE PROBLEMS

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 11

\*\*\* SYSTEM WARNING MESSAGE 3022

DATA BLOCK BGPOD IS REQUIRED AS INPUT AND IS NOT OUTPUT BY A PREVIOUS MODULE IN THE CURRENT DNAP ROUTE.

\*\*\* SYSTEM WARNING MESSAGE 3022

DATA BLOCK BGPOD IS REQUIRED AS INPUT AND IS NOT OUTPUT BY A PREVIOUS MODULE IN THE CURRENT DNAP ROUTE.

\*\*\* SYSTEM WARNING MESSAGE 3022

DATA BLOCK SIP IS REQUIRED AS INPUT AND IS NOT OUTPUT BY A PREVIOUS MODULE IN THE CURRENT DNAP ROUTE.

DEMONSTRATION PROBLEM - VIBRATION ANALYSIS  
CANTILEVER BEAM - SECOND OF THREE PROBLEMS

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 12

MODE NO.	EXTRACTION ORDER	EIGENVALUE	REAL EIGENVALUES		GENERALIZED MASS	GENERALIZED STIFFNESS
			RADIANS	CYCLES		
1	?	4.043853E+04	2.010933E+02	3.200500E+01	1.547996E-02	6.259870E+02
2	1	1.588253E+06	1.260259E+03	2.005765E+02	1.547818E-02	2.458327E+04
3	3	1.249620E+07	3.529335E+03	5.617110E+02	1.546320E-02	1.926127E+05

DEMONSTRATION PROBLEM - VIBRATION ANALYSIS  
CANTILEVER BEAM - SECOND OF THREE PROBLEMS

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 13

### E I G E N V A L U E   A N A L Y S I S   S U M M A R Y   ( I N V E R S E   P O W E R )

```

NUMBER OF EIGENVALUES EXTRACTED . . . . . 3
NUMBER OF STARTING POINTS USED . . . . . 1
NUMBER OF STARTING POINT MOVES . . . . . 0
NUMBER OF TRIANGULAR DECOMPOSITIONS . . . . . 3
TOTAL NUMBER OF VECTOR ITERATIONS . . . . . 24

REASON FOR TERMINATION . . . . . 6

LARGEST OFF-DIAGONAL MODAL MASS TERM . . . . 0.E+00
      . . . . . 0
MODE PAIR . . . . . 0
      . . . . . 0

NUMBER OF OFF-DIAGONAL MODAL MASS
TERMS FAILING CRITERION . . . . . 0
METHOD 1 NT,NBR PASSES = 1,EST. TIME = 00.0
METHOD 2 T,NBR PASSES = 1,EST. TIME = 00.0
    
```



# APPENDIX

DEMONSTRATION PROBLEM - VIBRATION ANALYSIS  
CANTILEVER BEAM - SECOND OF THREE PROBLEMS

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 14

EIGENVALUE = 4.043453E+04

REAL EIGENVECTOR NO. 1									
POINT ID.	TYPE	T1	T2	T3	R1	R2	R3		
1	G	0.0	0.0	0.0	0.0	0.0	0.0		
2	G	0.0	1.174160E-02	0.0	0.0	0.0	2.301641E-01		
3	G	0.0	4.510037E-02	0.0	0.0	0.0	4.32350E-01		
4	G	0.0	9.728581E-02	0.0	0.0	0.0	6.067422E-01		
5	G	0.0	1.655962E-01	0.0	0.0	0.0	7.537502E-01		
6	G	0.0	2.471493E-01	0.0	0.0	0.0	8.741623E-01		
7	G	0.0	3.395231E-01	0.0	0.0	0.0	9.692089E-01		
8	G	0.0	4.402039E-01	0.0	0.0	0.0	1.040638E+00		
9	G	0.0	5.469411E-01	0.0	0.0	0.0	1.090776E+00		
10	G	0.0	6.577473E-01	0.0	0.0	0.0	1.122560E+00		
11	G	0.0	7.709616E-01	0.0	0.0	0.0	1.139601E+00		
12	G	0.0	8.953159E-01	0.0	0.0	0.0	1.146081E+00		
13	G	0.0	1.000000E+00	0.0	0.0	0.0	1.147227E+00		

DEMONSTRATION PROBLEM - VIBRATION ANALYSIS  
CANTILEVER BEAM - SECOND OF THREE PROBLEMS

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 15

EIGENVALUE = 1.544253E+06

REAL EIGENVECTOR NO. 2									
POINT ID.	TYPE	T1	T2	T3	R1	R2	R3		
1	G	0.0	0.0	0.0	0.0	0.0	0.0		
2	G	0.0	-6.635232E-02	0.0	0.0	0.0	-1.225458E+00		
3	G	0.0	-2.250316E-01	0.0	0.0	0.0	-1.849124E+00		
4	G	0.0	-4.172538E-01	0.0	0.0	0.0	-1.905092E+00		
5	G	0.0	-5.896335E-01	0.0	0.0	0.0	-1.468560E+00		
6	G	0.0	-6.984782E-01	0.0	0.0	0.0	-6.574009E-01		
7	G	0.0	-7.136527E-01	0.0	0.0	0.0	3.775894E-01		
8	G	0.0	-6.210104E-01	0.0	0.0	0.0	1.471079E+00		
9	G	0.0	-4.276933E-01	0.0	0.0	0.0	2.466878E+00		
10	G	0.0	-1.349712E-01	0.0	0.0	0.0	3.242583E+00		
11	G	0.0	2.163380E-01	0.0	0.0	0.0	3.733730E+00		
12	G	0.0	6.025046E-01	0.0	0.0	0.0	3.946771E+00		
13	G	0.0	1.000000E+00	0.0	0.0	0.0	3.989046E+00		

DEMONSTRATION PROBLEM - VIBRATION ANALYSIS  
CANTILEVER BEAM - SECOND OF THREE PROBLEMS

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 16

EIGENVALUE = 1.245620E+07

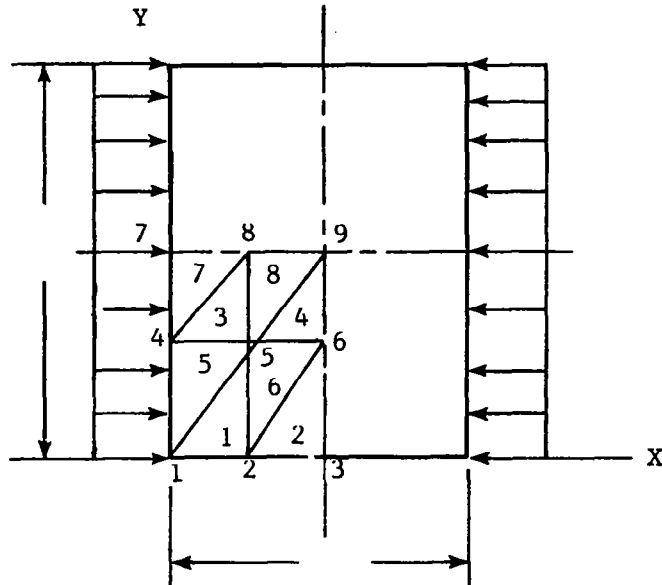
REAL EIGENVECTOR NO. 3									
POINT ID.	TYPE	T1	T2	T3	R1	R2	R3		
1	G	0.0	0.0	0.0	0.0	0.0	0.0		
2	G	0.0	1.675922E-01	0.0	0.0	0.0	2.885341E+00		
3	G	0.0	4.888639E-01	0.0	0.0	0.0	3.124550E+00		
4	G	0.0	7.243564E-01	0.0	0.0	0.0	1.319388E+00		
5	G	0.0	7.217143E-01	0.0	0.0	0.0	-1.416587E+00		
6	G	0.0	4.542449E-01	0.0	0.0	0.0	-3.756387E+00		
7	G	0.0	1.961643E-02	0.0	0.0	0.0	-4.620803E+00		
8	G	0.0	-4.062452E-01	0.0	0.0	0.0	-3.575105E+00		
9	G	0.0	-5.435493E-01	0.0	0.0	0.0	-9.736280E-01		
10	G	0.0	-5.812813E-01	0.0	0.0	0.0	2.218536E+00		
11	G	0.0	-2.174002E-01	0.0	0.0	0.0	4.883633E+00		
12	G	0.0	3.526910E-01	0.0	0.0	0.0	6.265297E+00		
13	G	0.0	1.000000E+00	0.0	0.0	0.0	6.576987E+00		

## APPENDIX

### Buckling Analysis

This problem is a buckling analysis of a simply supported rectangular plate under uniform compression.

Model



### Buckling Analysis DMAP Sequence

```

1. XDMAP      ERR=2,LIST,REF $   BUCKLING ANALYSIS
2. FILE      LAMA=APPEND/PHIA=APPEND $
3. GP1       GEOM1,GEOM2,/GPL,EQEXIN,GPDT,CSTM,BGPDT,SIL/S,N,LUSET/  V,N,
              NOGPDT $
4. GP2       GEOM2,EQEXIN/ECT $
5. PARAML    PCDB/**PRES**///NOPCDB $
6. PURGE     PLTSETX,PLTPAR,GPSETS,ELSETS/NOPCDB $
7. COND      P1,NOPCDB $
8. PLTSET    PCDB,EQEXIN,ECT/PLTSETX,PLTPAR,GPSETS,ELSETS/S,N,NSIL/S,N,
              JUMPPLOT=-1 $
9. PRTMSG    PLTSETX// $
10. PARAM    /**MPY*/PLTFLG/1/1 $
11. PARAM    /**MPY*/PFILE/0/0 $
12. COND     P1,JUMPPLOT $
13. PLOT     PLTPAR,GPSETS,ELSETS,CASECC,BGPDT,EQEXIN,SIL/
              PLOTX1/NSIL/LUSET/S,N,JUMPPLOT/S,N,PLTFLG/S,N,PFILE $
14. PRTMSG   PLOTX1// $
15. LABEL    P1 $
16. GP3      GEOM3,EQEXIN,GEOM2/SLT,  /S,N,NOGRAV $
17. PARAM    //C,N,AND/V,N,NOMGG/V,N,NOGRAV/V,Y,GRDPNT=-1 $
18. TAI      ECT,EPT,BGPDT,SIL,  ,CSTM/EST,GEI,GPECT,/V,N,LUSET/  S,N,
              NOSIMP/C,N,1/S,N,NOGENL/S,N,GENEL $
19. COND     ERROR1,NOSIMP $
20. PURGE    OGPST/GENEL $
21. PARAM    //C,N,ADD/V,N,NOKGGX/C,N,1/C,N,0 $

```

APPENDIX

22. EMG EST,CSTM,MPT,DIT,GEOM2,/KELM,KDICT,MELM,MDICT,,/S,N,NOKGGX/ S,  
N,NOMGG/C,N,/C,N,/C,N,/C,N,/C,Y,COUPMASS/C,Y,CPBAR/C,Y,CPROD/C,Y,  
CPQUAD1/C,Y,CPQUAD2/C,Y,CPTRIA1/C,Y,CPTRIA2/ C,Y,CPTUBE/C,Y,  
CPQDPLT/C,Y,CPTRPLT/C,Y,CPTRBSC \$

23. COND JMPKGG,NOKGGX \$

24. EMA GPECT,KDICT,KELM/KNN ,GPST \$

25. LABEL JMPKGG \$

26. COND JMPMGG,NOMGG \$

27. EMA GPECT,MDICT,MELM/MNN,/C,N,-1/C,Y,WTMASS=1.0 \$

28. LABEL JMPMGG \$

29. PARAM //C,N,MPY/V,N,NSKIP/C,N,0/C,N,0 \$

30. GP4 CASECC,GEOM4,EQEXIN,SIL,GPDT,BGPD,CSTM/RG,YS,USET,ASET/V,N,  
LUSET/S,N,MPCF1/S,N,MPCF2/S,N,SINGLE/S,N,OMIT/S,N,REACT/S,N,  
NSKIP/S,N,REPEAT/S,N,NOSET/S,N,NOL/S,N,NOA/C,Y,SUBID \$

31. COND ERROR6,NOL \$

32. PARAM //C,N,AND/V,N,NOSR/V,N,SINGLE/V,N,REACT \$

33. PURGE GM/MPCF1/GO,KOD,LOO,PO,UOOV,RUOV/OMIT/PS,KFS,KSS/SINGLE/ OG/  
NOSR \$

34. COND LBL4D,REACT \$

35. JUMP ERROR2 \$

36. LABEL LBL4D \$

37. GPSP GPL,GPST,USET,SIL/OGPST/S,N,NOGPST \$

38. COND LBL4,NOGPST \$

39. OFP OGPST,,,,// \$

40. LABEL LBL4 \$

41. EQUIV KNN,KFF/SINGLE \$

42. COND LBL3,SINGLE \$

43. SCE1 USET,KNN,,,/KFF,KFS,KSS,,, \$

44. LABEL LBL3 \$

45. EQUIV KFF,AAA/OMIT \$

46. COND LBL5,OMIT \$

47. SMP1 USET,KFF,,,/GO,AAA,KOD,LOO,,,,, \$

48. LABEL LBL5 \$

49. RBMG2 AAA/LLL \$

50. SSG1 SLT,BGPD,CSTM,SIL,EST,MPT, ,EDT,MNN,CASECC,DIT/PG/ V,N,  
LUSET/C,N,1 \$

51. EQUIV PG,PL/NOSET \$

52. COND LBL10,NOSET \$

53. SSG2 USET, ,YS,KFS,GO,,PG/,PO,PS,PL \$

54. LABEL LBL10 \$

55. SSG3 LLL,AAA,PL,LOO,KOD,PO/ULV,UOOV,RULV,RUOV/V,N,OMIT/V,Y,IRES=-1/  
C,N,1/S,N,EPSI \$

56. SDR1 USET,PG,ULV,UOOV,YS,GO,GM,PS,KFS,KSS,/UGV,PGG,OG/C,N,1/C,N,  
BKLO \$

57. SDR2 CASECC,CSTM,MPT,DIT,EQEXIN,SIL, ,EDT,BGPD,,OG,UGV,EST,,PGG/  
OPG1,OQG1,OUGV1,DES1,DEF1,PUGV1/C,N,BKLO \$

58. OFP OUGV1,OPG1,OQG1,DEF1,DES1,//S,N,CARDNO \$

59. COND P2,JUMPPLOT \$

60. PLOT PLTPAR,GPSETS,ELSETS,CASECC,BGPD,EQEXIN,SIL, PUGV1,,GPECT,  
DES1/PLOTX2/NSIL/LUSET/JUMPPLOT/PLTFLG/S,N,PFILE \$

61. PRTMSG PLOTX2// \$

62. LABEL P2 \$

63. TAI ECT,EPT,BGPD,SIL, ,CSTM/X1,X2,ECPT,GPCT/V,N,LUSET/ V,N,  
NOSIMP/C,N,0/V,N,NOGENL/V,N,GENEL \$

64. DSMG1 CASECC, ,SIL,EDT,UGV,CSTM,MPT,ECPT,GPCT,DIT/KDNN/ V,N,  
DSCASET \$

65. EQUIV KDNN,KDFF/SINGLE \$

66. COND LBL3D,SINGLE \$

## APPENDIX

```

67. SCE1      USET,KONN,,,/KOFF,KDFS,,,, $
68. LABEL    LBL3D $
69. EQUIV    KOFF,KDAA/OMIT $
70. COND     LBL5D,OMIT $
71. SMP2     USET,GO,KOFF/KDAA $
72. LABEL    LBL5D $
73. ADD      KDAA,/KDAAM/C,N,(-1.0,0.0)/C,N,(0.0,0.0) $
74. DPD      DYNAMICS,GPL,SIL,USET/GPLD,SILD,USED,,,,,,,,EED,EQDYN/V,N,
             LUSET/V,N,LUSETD/V,N,NOTFL/V,N,NODLT/V,N,NOPSDL/V,N,NOFRL/ V,
             N,NONLFT/V,N,NOTRL/S,N,NOEED/C,N,/V,N,NOUE $
75. COND     ERROR3,NOEED $
76. PARAM    //C,N,MPY/V,N,NEIGV/C,N,1/C,N,-1 $
77. READ     KAA,KDAAM,,,EED,USET,CASECC/LAMA,PHIA,,OEIGS/C,N,BUCKLING/ S,N,
             NEIGV/C,N,2 $
78. OFF      OEIGS,LAMA,,,,//S,N,CARDNO $
79. COND     ERROR4,NEIGV $
80. SDR1     USET,,PHIA,,,GO,GM,,,KFS,,/PHIG,,BQG/C,N,1/C,N,BKL1 $
81. SDR2     CASECC,CSTM,MPT,DIT,EQEXIN,SIL,,,BGPDT,LAMA,BQG,PHIG,EST,,/,
             OBQG1,OPHIG,OBES1,OBEF1,PPHIG/C,N,BKL1 $
82. OFF      OPHIG,OBQG1,OBEF1,OBES1,,,/V,N,CARDNO $
83. COND     P3,JUMPPLOT $
84. PLOT     PLTPAR,GPSETS,ELSETS,CASECC,BGPDT,EQEXIN,SIL,,PPHIG,GPECT,
             OBES1/PLOTX3/NSIL/LUSET/JUMPPLOT/PLTFLG/S,N,PFILE $
85. PRTMSG   PLOTX3// $
86. LABEL    P3 $
87. JUMP     FINIS $
88. LABEL    ERROR1 $
89. PRTPARM  //C,N,-1/C,N,BUCKLING $
90. LABEL    ERROR2 $
91. PRTPARM  //C,N,-2/C,N,BUCKLING $
92. LABEL    ERROR3 $
93. PRTPARM  //C,N,-3/C,N,BUCKLING $
94. LABEL    ERROR4 $
95. PRTPARM  //C,N,-4/C,N,BUCKLING $
96. LABEL    ERROR6 $
97. PRTPARM  //C,N,-6/C,N,BUCKLING $
98. LABEL    FINIS $
99. END      $

```

### Description of DMAP Operations for Buckling Analysis

3. GP1 generates coordinate system transformation matrices, tables of grid-point locations, and tables for relating internal and external grid-point numbers.
4. GP2 generates an element connection table with internal indices.
7. Go to DMAP no. 15 if no plot package is present.
8. PLTSET transforms user input into a form used to drive structure plotter.
9. PRTMSG prints error messages associated with structure plotter.
12. Go to DMAP no. 15 if no undeformed structure plot request.

## APPENDIX

13. PLOT generates all requested undeformed structure plots.
14. PRTMSG prints plotter data and engineering data for each undeformed plot generated.
16. GP3 generates static loads table.
18. TAL generates element tables for use in matrix assembly and stress recovery.
19. Go to DMAP no. 88 and print error message if no structural elements.
22. EMG generates structural element matrix tables and dictionaries for later assembly.
23. Go to DMAP no. 25 if no stiffness matrix is to be assembled.
24. EMA assembles stiffness matrix  $[K_{gg}^x]$  and grid-point singularity table.
26. Go to DMAP no. 28 if no mass matrix is to be assembled.
27. EMA assembles mass matrix  $[M_{gg}]$ .
30. GP4 generates flags defining members of various displacement sets (USET) and forms enforced displacement vector  $\{Y_s\}$ .
31. Go to DMAP no. 56 and print error message if no independent degrees of freedom are defined.
34. Go to DMAP no. 56 if no free-body supports supplied.
35. Go to DMAP no. 90 and print error message if free-body supports are present.
37. GPSP determines if possible grid-point singularities remain.
38. Go to DMAP no. 40 if no grid-point singularity table.
40. Equivalence  $[K_{nn}]$  to  $[K_{ff}]$  if no single-point constraints.
41. Go to DMAP no. 44 if no single-point constraints.
43. SCE1 partitions out single-point constraints

$$[K_{nn}] = \begin{bmatrix} K_{ff} & K_{fs} \\ K_{sf} & K_{ss} \end{bmatrix}$$

45. Equivalence  $[K_{ff}]$  to  $[K_{aa}]$  if no omitted coordinates.
46. Go to DMAP no. 48 if no omitted coordinates.

APPENDIX

47. SMP1 partitions constrained stiffness matrix

$$[K_{ff}] = \begin{bmatrix} \bar{K}_{aa} & | & K_{ao} \\ \hline K_{oa} & | & K_{oo} \end{bmatrix}$$

solves for transformation matrix  $[G_o] = -[K_{oo}]^{-1}[K_{oa}]$ , and performs matrix reduction  $[K_{aa}] = [\bar{K}_{aa}] + [K_{oa}^T][G_o]$ .

49. RBMG2 decomposes constrained stiffness matrix  $[K_{aa}] = [L_{\ell\ell}][U_{\ell\ell}]$ .

50. SSG1 generates static load vectors  $\{P_g\}$ .

51. Equivalence  $\{P_g\}$  to  $\{P_\ell\}$  if no restraints applied.

52. Go to DMAP no. 54 if no constraints applied.

53. SSG2 applies constraints to static load vectors

$$\{P_g\} = \begin{Bmatrix} \bar{P}_n \\ - \\ P_m \end{Bmatrix} \quad \{P_n\} = \{\bar{P}_n\} + [G_m^T] \{P_m\}$$

$$\{P_n\} = \begin{Bmatrix} \bar{P}_f \\ - \\ P_s \end{Bmatrix} \quad \{P_f\} = \{\bar{P}_f\} - [K_{fs}] \{P_s\}$$

$$\{P_f\} = \begin{Bmatrix} \bar{P}_a \\ - \\ P_o \end{Bmatrix} \quad \text{and} \quad \{P_\ell\} = \{\bar{P}_a\} + [G_o^T] \{P_o\}$$

55. SSG3 solves for displacements of independent coordinates

$$\{u_\ell\} = [K_{\ell\ell}]^{-1} \{P_\ell\}$$

solves for displacements of omitted coordinates

$$\{u_o^o\} = [K_{oo}]^{-1} \{P_o\}$$

calculates residual vector (RULV) and residual vector error ratio for independent coordinates

APPENDIX

$$\{\delta P_\ell\} = \{P_\ell\} - [K_{\ell\ell}] \{u_\ell\}$$

$$\epsilon_\ell = \frac{\begin{Bmatrix} u_\ell^T \\ \end{Bmatrix} \{\delta P_\ell\}}{\begin{Bmatrix} P_\ell^T \\ \end{Bmatrix} \{u_\ell\}}$$

and calculates residual vector (RUØV) and residual vector error ratio for omitted coordinates

$$\{\delta P_o\} = \{P_o\} - [K_{oo}] \{u_o^o\}$$

$$\epsilon_o = \frac{\begin{Bmatrix} u_o^T \\ \end{Bmatrix} \{\delta P_o\}}{\begin{Bmatrix} P_o^T \\ \end{Bmatrix} \{u_o\}}$$

56. SDR1 recovers dependent displacements

$$\{u_o\} = [G_o] \{u_\ell\} + \{u_o^o\}$$

$$\begin{Bmatrix} u_a \\ - \\ u_o \end{Bmatrix} = \{u_f\} \qquad \begin{Bmatrix} u_f \\ - \\ y_s \end{Bmatrix} = \{u_n\}$$

$$\{u_m\} = [G_m] \{u_n\} \qquad \begin{Bmatrix} u_n \\ - \\ u_m \end{Bmatrix} = \{u_g\}$$

and recovers single-point forces of constraint

$$\{q_s\} = -\{P_s\} + [K_{fs}^T] \{u_f\} + [K_{ss}] \{y_s\}$$

57. SDR2 calculates element forces and stresses (ØEF1, ØES1) and prepares load vectors, displacement vectors, and single-point forces of constraint for output (ØPG1, ØUGV1, PUGV1, ØØG1).

58. ØFP formats tables prepared by SDR2 and places them on the system output file for printing.

59. Go to DMAP no. 62 if no static deformed structure plots are requested.

60. PLOT generates all requested static deformed structure plots.

APPENDIX

61. PRTMSG prints plotter data and engineering data for each deformed plot generated.
63. TAL generates element tables for use in differential stiffness matrix assembly.
64. DSMG1 generates differential stiffness matrix  $[K_{gg}^d]$ .
65. Equivalence  $[K_{nn}^d]$  to  $[K_{ff}^d]$  if no single-point constraints.
66. Go to DMAP no. 68 if no single-point constraints.
67. SCEL partitions out single-point constraints

$$[K_{nn}^d] = \left[ \begin{array}{c|c} K_{ff}^d & K_{fs}^d \\ \hline K_{sf}^d & K_{ss}^d \end{array} \right]$$

69. Equivalence  $[K_{ff}^d]$  to  $[K_{aa}^d]$  if no omitted coordinates.
70. Go to DMAP no. 72 if no omitted coordinates.
71. SMP2 partitions constrained differential stiffness matrix

$$[K_{ff}^d] = \left[ \begin{array}{c|c} \bar{K}_{aa}^d & K_{ao}^d \\ \hline K_{oa}^d & K_{oo}^d \end{array} \right]$$

and performs matrix reduction

$$[K_{aa}^d] = [\bar{K}_{aa}^d] + [K_{oa}^d]^T [G_o] + [G_o]^T [K_{oa}^d] + [G_o]^T [K_{oo}^d] [G_o]$$

74. DPD extracts eigenvalue extraction data from dynamics data block.
75. Go to DMAP no. 92 and print error message if no eigenvalue extraction data.
77. READ extracts real eigenvalues from the equation

$$[K_{\ell\ell} + \lambda K_{\ell\ell}^d] \{u_\ell\} = 0$$

and normalizes eigenvectors according to one of the following user requests:

Unit value of selected coordinate

Unit value of largest component



APPENDIX

78. ØFP formats eigenvalues and summary of eigenvalue extraction information and places them on the system output file for printing.
79. Go to DMAP no. 94 and print error message if no eigenvalues found.
80. SDR1 recovers dependent components of the eigenvectors

$$\begin{aligned} \{\phi_o\} &= [G_o] \{\phi_a\} & \begin{Bmatrix} \phi_a \\ - \\ \phi_a \end{Bmatrix} &= \{\phi_f\} \\ \\ \begin{Bmatrix} \phi_f \\ - \\ \phi_s \end{Bmatrix} &= \{\phi_n\} & \{\phi_m\} &= [G_m] \{\phi_n\} \\ \\ \begin{Bmatrix} \phi_n \\ - \\ \phi_m \end{Bmatrix} &= \{\phi_g\} \end{aligned}$$

and recovers single-point forces of constraint  $\{q_s\} = [K_{fs}^T] \{\phi_f\}$ .

81. SDR2 calculates element forces and stresses (ØBEF1, ØBES1) and prepares eigenvectors and single-point forces of constraint for output (ØPHIG, PPHIG, ØBQGL).
82. ØFP formats tables prepared by SDR2 and places them on the system output file for printing.
83. Go to DMAP no. 86 if no deformed (buckling) structure plots are requested.
84. PLØT generates all requested deformed (buckling) structure plots.
85. PRTMSG prints plotter data and engineering data for each deformed plot generated.
87. Go to DMAP no. 98 and make normal exit.
89. BUCKLING ANALYSIS ERROR MESSAGE NØ. 1 - NØ STRUCTURAL ELEMENTS HAVE BEEN DEFINED.
91. BUCKLING ANALYSIS ERROR MESSAGE NØ. 2 - FREE-BØDY SUPPORTS NØT ALLOWED.
93. BUCKLING ANALYSIS ERROR MESSAGE NØ. 3 - EIGENVALUE EXTRACTION DATA REQUIRED FØR REAL EIGENVALUE ANALYSIS.
95. BUCKLING ANALYSIS ERROR MESSAGE NØ. 4 - NØ EIGENVALUES FØUND.
97. BUCKLING ANALYSIS ERROR MESSAGE NØ. 6 - NØ INDEPENDENT DEGREES ØF FREEDØM HAVE BEEN DEFINED.

# APPENDIX

## Buckling Analysis Sample Problem

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 1

N A S T R A N   E X E C U T I V E   C O N T R O L   D E C K   E C H O

```
TO HINIYRAN,BUCKLING
APP DMAP
XDMAP ERR=2,LIST,REF $ BUCKLING ANALYSIS
(SEE NASTRAN SOURCE PROGRAM COMPILATION FOR LISTING OF DMAP SEQUENCE)
END $
TIME 5
CEND
```

BUCKLING OF A SIMPLY-SUPPORTED RECTANGULAR PLATE UNDER UNIFORM COMPRESSION      JANUARY 25, 1977      NASTRAN 1/ 1/77      PAGE 2

C A S E   C O N T R O L   D E C K   E C H O

```
CARD
COUNT
1      TITLE = BUCKLING OF A SIMPLY-SUPPORTED RECTANGULAR PLATE UNDER UNIFORM
2      SUBTITLE = COMPRESSION
3      SPC = 1
4      DISP = ALL
5      SUBCASE 1
6      LOAD = 10
7      SUBCASE 2
8      METHOD = 45
9      PLOTID TEST PLOT CAPABILITY
10     OUTPUT(PLOT)
11     SET 1 INCLUDE 1 THRU 8
12     PLOTTER CALCOMP.765
13     MAXIMUM OFFFORMATION 3.
14     FIND SCALE,ORIGIN 1,SET 1
15     $
16     $PLOT UNDEFORMED SHAPE
17     $
18     PLOT
19     $
20     $ PLOT STATIC DEFORMATION IN SUBCASE 1
21     $
22     PLOT STATIC DEFORMATION 1
23     $
24     $ PLOT MODAL DEFORMATION IN SUBCASE 2
25     $
26     PLOT MODAL DEFORMATION 2
27     BEGIN BULK
```

\*\*\* USER INFORMATION MESSAGE 207. BULK DATA NOT SORTED,KSORT WILL RE-ORDER DECK.

# APPENDIX

CARD COUNT	1	2	3	4	5	6	7	8	9	10
1-	CTRIA2	1	20	1	2	5				
2-	CTRIA2	2	20	2	3	6				
3-	CTRIA2	3	20	4	5	8				
4-	CTRIA2	4	20	5	6	9				
5-	CTRIA2	5	20	5	4	1				
6-	CTRIA2	6	20	6	5	2				
7-	CTRIA2	7	20	8	7	4				
8-	CTRIA2	8	20	9	8	5				
9-	EIGB	45	INV	.0	1.0	1	1	0		+A
10-	+A	MAX								
11-	FORCE	10	1		1.0	1.0	.0	.0		
12-	FORCE	10	4		1.0	1.0	.0	.0		
13-	FORCE	10	7		1.0	1.0	.0	.0		
14-	GRID	1		.0	.0					
15-	GRID	2		2.4	.0					
16-	GRID	3		4.8	.0					
17-	GRID	4		.0	3.0					
18-	GRID	5		2.4	3.0					
19-	GRID	6		4.8	3.0					
20-	GRID	7		.0	6.0					
21-	GRID	8		2.4	6.0					
22-	GRID	9		4.8	6.0					
23-	MAT1	30	3.0*7		.3					
24-	PTRIA2	20	30	.096						
25-	SPC	1	1	3456						
26-	SPC	1	2	356						
27-	SPC	1	3	1356						
28-	SPC	1	4	346						
29-	SPC	1	5	6						
30-	SPC	1	6	156						
31-	SPC	1	7	2346						
32-	SPC	1	8	245						
33-	SPC	1	9	12456						
	ENDDATA									

\*\*\* USER POTENTIALLY FATAL MESSAGE 22.  
 POSSIBLE ERROR IN DMAP INSTRUCTION SDR1 INSTRUCTION NO. 57  
 DATA BLOCK NAMED GM APPEARS AS INPUT BEFORE BEING DEFINED  
 \*\*NO ERRORS FOUND - EXECUTE NASTRAN PROGRAM\*\*

MESSAGES FROM THE PLOT MODULE

P L O T T E R   D A T A

THE FOLLOWING PLOTS ARE FOR A CALCOMP 765 INCREMENTAL PLOTTER (2 CHARACTERS/COMMAND - .005 STEP SIZE)  
 AN END-OF-FILE MARK FOLLOWS THE LAST PLOT  
 THE FIRST COMMAND FOR EACH PLOT CONTAINS THE PLOT NUMBER  
 PEN 1 - SIZE 1, BLACK

E N G I N E E R I N G   D A T A

ORTHOGRAPHIC PROJECTION  
 ROTATIONS (DEGREES) - GAMMA = 34.27, BETA = 23.17, ALPHA = 0.00, AXES = +X,+Y,+Z, SYMMETRIC  
 SCALE (OBJECT-TO-PLOT SIZE) = 6.248091E-01  
 ORIGIN      1      -      X0 = -3.246604E+00, Y0 = -5.732292E+00      (INCHES)

# APPENDIX

BUCKLING OF A SIMPLY-SUPPORTED RECTANGULAR PLATE UNDER UNIFORM COMPRESSION      JANUARY 25, 1977    NASTRAN 1/ 1/77    PAGE 11

MESSAGES FROM THE PLOT MODULE

PLOT 1    UNDEFORMED SHAPE

```

*** SYSTEM INFORMATION MESSAGE 3113, EMGPRO PROCESSING SINGLE PRECISION ELEMENTS OF TYPE 17 STARTING WITH ID 1
*** SYSTEM INFORMATION MESSAGE 3107, EMGOLD IS PROCESSING ELEMENTS OF TYPE = 17, BEGINNING WITH ELEMENT ID = 1
***USER INFORMATION MESSAGE 3023--PARAMETERS FOR SYMMETRIC DECOMPOSITION OF DATA BLOCK KAA ( N = 24 )
      TIME ESTIMATE= 1            C AVG = 8            PC AVG = 0    SPILL GROUPS = 0            S AVG = 1
      ADDITIONAL CORE= -21933      C MAX = 14            PCHAX = 0    PC GROUPS = 0            PREFACE LOOPS = 1
*** SYSTEM WARNING MESSAGE 2363, SSG2B FORCED MPYAD COMPATIBILITY OF MATRIX ON 103, FROM ( 30, 2), TO ( 30, 1)
      MPYAD--NULL MATRIX PRODUCT
      METHOD 1 NT,NBR PASSES = 1,EST. TIME = 00.0
  
```

BUCKLING OF A SIMPLY-SUPPORTED RECTANGULAR PLATE UNDER UNIFORM COMPRESSION      JANUARY 25, 1977    NASTRAN 1/ 1/77    PAGE 12

\*\*\* USER INFORMATION MESSAGE 3035

FOR LOAD 1 EPSILON SUB E = -4.6875070E-14

MPYAD--NULL MATRIX PRODUCT  
METHOD 2 T ,NBR PASSES = 1,EST. TIME = 00.0

BUCKLING OF A SIMPLY-SUPPORTED RECTANGULAR PLATE UNDER UNIFORM COMPRESSION      JANUARY 25, 1977    NASTRAN 1/ 1/77    PAGE 13

SUBCASE 1

D I S P L A C E M E N T   V E C T O R

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
1	G	9.829562E-07	-3.572685E-07	0.0	0.0	0.0	0.0
2	G	4.814494E-07	-2.975060E-07	0.0	0.0	0.0	0.0
3	G	0.0	-2.977195E-07	0.0	0.0	0.0	0.0
4	G	7.847110E-07	-2.163574E-07	0.0	0.0	0.0	0.0
5	G	3.708394E-07	-1.417374E-07	0.0	0.0	0.0	0.0
6	G	0.0	-1.172926E-07	0.0	0.0	0.0	0.0
7	G	9.409551E-07	0.0	0.0	0.0	0.0	0.0
8	G	4.363925E-07	0.0	0.0	0.0	0.0	0.0
9	G	0.0	0.0	0.0	0.0	0.0	0.0

# APPENDIX

BUCKLING OF A SIMPLY-SUPPORTED RECTANGULAR PLATE UNDER UNIFORM  
COMPRESSION

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 14

## MESSAGES FROM THE PLOT MODULE

### P L O T T E R   D A T A

THE FOLLOWING PLOTS ARE FOR A CALCOMP 765 INCREMENTAL PLOTTER (2 CHARACTERS/COMMAND - .005 STEP SIZE)

AN END-OF-FILE MARK FOLLOWS THE LAST PLOT

THE FIRST COMMAND FOR EACH PLOT CONTAINS THE PLOT NUMBER

PEN 1 - SIZE 1, BLACK

### E N G I N E E R I N G   D A T A

ORTHOGRAPHIC PROJECTION  
ROTATIONS (DEGREES) - GAMMA = 34.27, BETA = 23.17, ALPHA = 0.00, AXES = +X,+Y,+Z, SYMMETRIC  
SCALE (OBJECT-TO-PLOT SIZE) = 6.248091E-01

ORIGIN        1        -        X0 = -3.246604E+00, Y0 = -5.732292E+00        (INCHES)

BUCKLING OF A SIMPLY-SUPPORTED RECTANGULAR PLATE UNDER UNIFORM  
COMPRESSION

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 15

## MESSAGES FROM THE PLOT MODULE

PLOT    2    STATIC DEFORM.        1 - SUBCASE        10 - LOAD

\*\*\*USER INFORMATION MESSAGE 3023--PARAMETERS FOR SYMMETRIC DECOMPOSITION OF DATA BLOCK LAHA        ( N =    24 )  
      TIME ESTIMATE=        1        C AVG =        8        PC AVG =        0        SPILL GROUPS =        0        S AVG =        1  
      ADDITIONAL CORE= -19710        C MAX =        14        PCMAX =        0        PC GROUPS =        0        PREFACE LOOPS =        1

BUCKLING OF A SIMPLY-SUPPORTED RECTANGULAR PLATE UNDER UNIFORM  
COMPRESSION

JANUARY 25, 1977 NASTRAN 1/ 1/77 PAGE 16

## E I G E N V A L U E   A N A L Y S I S   S U M M A R Y        (INVERSE POWER)

```

NUMBER OF EIGENVALUES EXTRACTED . . . . . 1
NUMBER OF STARTING POINTS USED . . . . . 1
NUMBER OF STARTING POINT MOVES . . . . . 0
NUMBER OF TRIANGULAR DECOMPOSITIONS . . . . . 1
TOTAL NUMBER OF VECTOR ITERATIONS . . . . . 6
REASON FOR TERMINATION . . . . . 6
LARGEST OFF-DIAGONAL MODAL MASS TERM . . . . 0.E+00
      MODE PAIR . . . . . 0
      . . . . . 0
NUMBER OF OFF-DIAGONAL MODAL MASS
TERMS FAILING CRITERION . . . . . 0
    
```

# APPENDIX

BUCKLING OF A SIMPLY-SUPPORTED RECTANGULAR PLATE UNDER UNIFORM COMPRESSION      JANUARY 25, 1977      NASTRAN 1/ 1/77      PAGE 17

MODE NO.	EXTRACTION ORDER	EIGENVALUE	REAL EIGENVALUES		GENERALIZED MASS	GENERALIZED STIFFNESS
			RADIANS	CYCLES		
1	1	1.489206E+03	3.859023E+01	6.141825E+00	0.0	0.0
			METHOD 2 T ,NBR PASSES =		1,EST. TIME =	00.0

BUCKLING OF A SIMPLY-SUPPORTED RECTANGULAR PLATE UNDER UNIFORM COMPRESSION      JANUARY 25, 1977      NASTRAN 1/ 1/77      PAGE 18

SUBCASE 2

EIGENVALUE = 1.489206E+03

		REAL EIGENVECTOR NO.				1		
POINT ID.	TYPE	T1	T2	T3	R1	R2	R3	
1	G	-1.305304E-27	2.814161E-27	0.0	0.0	0.0	0.0	
2	G	-1.122789E-27	-1.044400E-27	0.0	1.895820E-01	0.0	0.0	
3	G	0.0	-5.651426E-28	0.0	2.544363E-01	0.0	0.0	
4	G	4.380746E-28	1.662038E-27	0.0	0.0	-2.580528E-01	0.0	
5	G	3.911387E-28	-5.317506E-28	5.064851E-01	1.330569E-01	-1.743449E-01	0.0	
6	G	0.0	-9.803011E-28	7.040344E-01	1.893988E-01	0.0	0.0	
7	G	3.010605E-28	0.0	0.0	0.0	-3.458518E-01	0.0	
8	G	7.109578E-28	0.0	7.113585E-01	0.0	-2.463882E-01	0.0	
9	G	0.0	0.0	1.000000E+00	0.0	0.0	0.0	

BUCKLING OF A SIMPLY-SUPPORTED RECTANGULAR PLATE UNDER UNIFORM COMPRESSION      JANUARY 25, 1977      NASTRAN 1/ 1/77      PAGE 19

MESSAGES FROM THE PLOT MODULE

P L O T T E R   D A T A

THE FOLLOWING PLOTS ARE FOR A CALCOMP 765 INCREMENTAL PLOTTER (2 CHARACTERS/COMMAND - .005 STEP SIZE)

AN END-OF-FILE MARK FOLLOWS THE LAST PLOT

THE FIRST COMMAND FOR EACH PLOT CONTAINS THE PLOT NUMBER

PEN 1 - SIZE 1, BLACK

E N G I N E E R I N G   D A T A

ORTHOGRAPHIC PROJECTION  
 ROTATIONS (DEGREES) - GAMMA = 34.27, BETA = 23.17, ALPHA = 0.00, AXES = +X,+Y,+Z, SYMMETRIC  
 SCALE (OBJECT-TO-PLOT SIZE) = 6.248091E-01

ORIGIN      1      -      X0 = -3.246604E+00, Y0 = -5.732292E+00      (INCHES)

BUCKLING OF A SIMPLY-SUPPORTED RECTANGULAR PLATE UNDER UNIFORM COMPRESSION      JANUARY 25, 1977      NASTRAN 1/ 1/77      PAGE 20

MESSAGES FROM THE PLOT MODULE

PLOT 3 MODAL DEFORM.      2 - SUBCASE      1 - MODE      1.489206E+03 - EIGENVALUE

APPENDIX

Outputs From the Plot Module

Figures A1 to A3 are outputs from the plot module, and they show various stages in the buckling of a simply supported rectangular plate under uniform compression.

1 1/31/77

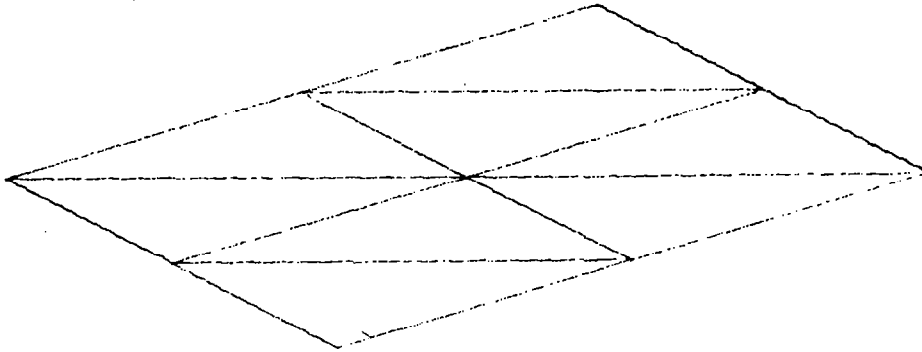


Figure A1.- Buckling of a simply supported rectangular plate under uniform compression; undeformed shape.

2 1/31/77 Maximum deformation = 9.82956-07

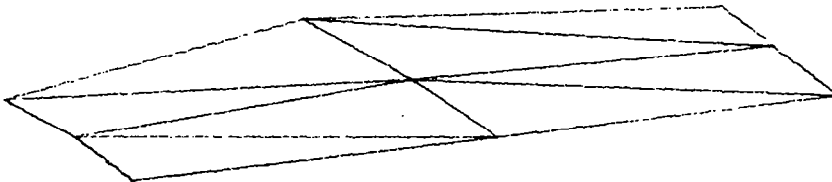


Figure A2.- Buckling of a simply supported rectangular plate under uniform compression; static deformation; subcase 1, load 10.

APPENDIX

3 1/31/77 Maximum deformation = 1.00000000

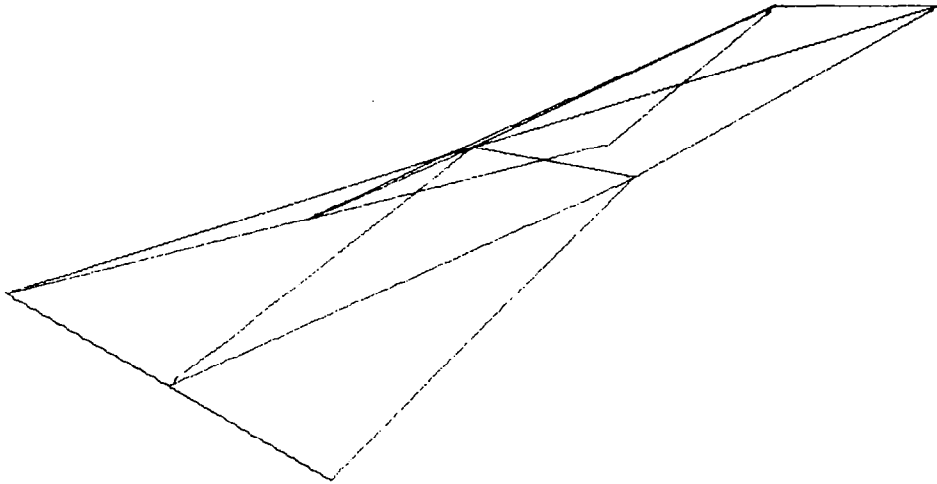


Figure A3.- Buckling of a simply supported rectangular plate under uniform compression; modal deformation; subcase 2; mode 1.

Matrix Operation and Data Block Manipulation

This problem shows how to use the matrix operations and manipulate data blocks.

Model

No model.

The following matrices are input by means of DMI bulk data card:

$$[A] = \begin{bmatrix} 1. & .5 & .33333 \\ .5 & .33333 & .25 \\ .33333 & .25 & .2 \end{bmatrix}$$

$$[B] = \begin{bmatrix} 1. \\ 2. \\ 3. \end{bmatrix}$$

$$[CP] = \begin{bmatrix} 1. \\ 1. \\ 0. \end{bmatrix}$$

$$[RP] = \begin{bmatrix} 0. \\ 1. \\ 1. \end{bmatrix}$$



## APPENDIX

### Matrix Operation and Data Block Manipulation DMAP Sequence

```

1. XDMAP      ERR=2,LIST,REF $   MATRIX AND DATA BLOCK MANIPULATIONS
2. DECOMP    A/L,U///MINDIAG/DET/POWER/SING $
3. FBS       L,U,B/X/1 $
4. MATPRN    A,L,U,B,X // $
5. PARTN     A,CP,RP/A11,A21,A12,A22/ $
6. MERGE     A11,A21,A12,A22,CP,RP/Y/ $
7. MATPRN    Y,CP,RP// $
8. MATPRN    A11,A21,A12,A22 $
9. MPYAD     A,B/Z/O/ $
10. SOLVE    A,B/W/ $   W SHOULD EQUAL X
11. MATPRN    X,W,Z $
12. TRNSP    A/V $   V SHOULD EQUAL A(SYMMETRY)
13. DIAGONAL A/T//O.$ GET IDENTITY MATRIX
14. SCALAR   T $
15. MATPRN    V,T// $
16. OUTPUT2  A,B// -1/11/ $
17. INPUTT2  /C,D/ -1/11 $ A=C , B=D
18. MATPRN    C,D// $
19. SETVAL   //S,N,ISW/ -1 $
20. COPY     A/E/ISW $
21. SWITCH   A,B//ISW $
22. SWITCH   A,B//ISW $
23. MATPRN    E,A,B// $
24. PARAMR   /**ADD*/X2/O./O. $
25. PARAML   A/**DMI*///V,N,X1/ $
26. LABEL    LOOP $
27. PARAMR   /**ADD*/X1/X1/1. $
28. PARAMR   /**MPY*/X2/X2/2. $
29. PRTPARM  //O $
30. EXIT     3 $
31. REPT     LOOP,NLOOP $
32. END      $

```

### Description of DMAP Sequence for Matrix Operations and Data Block Manipulations

2. DECOMP decomposes square matrix  $[A]$  into upper  $[U]$  and lower  $[L]$  triangular factors.
3. FBS solves the matrix equation  $[L][U][X] = [B]$ .
5. PARTN partitions  $[A]$  into  $[A11]$ ,  $[A12]$ ,  $[A21]$ , and  $[A22]$ .
6. MERGE forms the matrix  $[Y]$  from the partitions in step 5;  $[A] = [Y]$ .
9. MPYAD performs matrix multiplication  $[A][B] = [Z]$ .
10. SOLVE solves the matrix equation  $[A][W] = [B]$ .  $[W] = [X]$  from step 3.
12. TRNSP forms  $[A]^T$  given  $[A]$ .  $[A]^T = [U] = [A]$  because of symmetry.
13. DIAGONAL removes the real part of the diagonal from matrix  $[A]$ , raises each term to the 0 (zero) power, and outputs the identity matrix  $[I]$ .

APPENDIX

14. SCALAR extracts element from row 1, column 1 of [T].
16. OUTPUT2 writes data blocks [A] and [B] onto unit 11.
17. INPUTT2 reads data blocks [C] and [D] from unit 11. [A] = [C], [B] = [D].
19. SETVAL sets DMAP parameter variable ISW = -1.
20. COPY copies data block [A] onto [E].
21. SWITCH interchanges data block names for [A] and [B].
22. SWITCH interchanges data block names for [A] and [B] back to original names.
24. PARAMR adds 0. + 0. and stores in parameter X2.
25. PARMAL selects parameter 1. from [A] and stress in X1.
26. LABEL starts a loop.
27. PARAMR adds 1 to X1 and stores back into X1.
28. PARAMR multiplies X2 by 2 and stores back into X2.
29. PRTPARM prints all parameters.
30. EXIT will conditionally terminate job after the EXIT instruction has been ignored 3 times.
31. REPT repeats group of DMAP instructions starting at step 26 NLOOP times. NLOOP set by a parameter.

Matrix Operation Sample Problem

JANUARY 24, 1977 NASTRAN 1/ 1/77 PAGE 1

N A S T R A N   F X E C U T I V E   C O N T R O L   D E C K   E C H O

```

ID      MINITRAN,DEMO-4
APP     DMAP
XOMAP  ERR=2,LIST,REF $
(SEE NASTRAN SOURCE PROGRAM COMPILATION FOR LISTING OF DMAP SEQUENCE)
END $
TIME    3
CEND
    
```

JANUARY 24, 1977 NASTRAN 1/ 1/77 PAGE 2

C A S E   C O N T R O L   D E C K   E C H O

```

CAPD
COUNT
1      BEGIN BULK
    
```

APPENDIX

S O R T E D   S U L K   D A T A   E C H O .

CARD COUNT	1	2	3	4	5	6	7	8	9	10
1-	DMI A	0	6	1	1					
2-	DMI A	1	1	1	.5	.33333	.33333			
3-	DMI A	2	1	.5	.33333	.25				
4-	DMI A	3	1	.33333	.25	.2				
5-	DMI B	0	2	1	1		3		1	
6-	DMI B	1	1	1	2	3				
7-	DMI CP	0	2	1	1		3		1	
8-	DMI CP	1	1	1	1	0.0				
9-	DMI RP	0	2	1	1		3		1	
10-	DMI RP	1	1	0.0	1	1				
11-	PARAM NLOOP	4								

ENDDATA

\*\*NO ERRORS FOUND - EXECUTE NASTRAN PROGRAM\*\*

\*\*\*USER INFORMATION MESSAGE 3023--PARAMETERS FOR SYMMETRIC DECOMPOSITION OF DATA BLOCK A ( N = 3 )  
 TIME ESTIMATE = 1 C AVG = 2 PC AVG = 0 SPILL GROUPS = 0 S AVG = 1  
 ADDITIONAL CORE = -15660 C MAX = 3 PCHAX = 0 PC GROUPS = 0 PREFACE LOOPS = 1

MATRIX A (GINO NAME 101 ) IS A REAL 3 COLUMN X 3 ROW SYMETRIC MATRIX.

COLUMN	1	ROWS	1 THRU	3
1.00000E+00	5.00000E-01	3.33330E-01		
5.00000E-01	3.33330E-01	2.50000E-01		
3.33330E-01	2.50000E-01	2.00000E-01		

THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3  
 THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.

MATRIX L (GINO NAME 102 ) IS A REAL 3 COLUMN X 3 ROW LOW TRI MATRIX.

COLUMN	1	ROWS	1 THRU	3
1.00000E+00	-5.00000E-01	-3.33330E-01		
8.33300E-02	-1.00006E+00			
5.55111E-03				

THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3  
 THE DENSITY OF THIS MATRIX IS 66.66 PERCENT.

APPENDIX

MATRIX B (GINO NAME 104 ) IS A REAL 1 COLUMN X 3 ROW RECTANG MATRIX.  
 COLUMN 1 ROWS 1 THRU 3 -----  
 1.00000E+00 2.00000E+00 3.00000E+00  
 THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3  
 THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.

MATRIX X (GINO NAME 105 ) IS A REAL 1 COLUMN X 3 ROW RECTANG MATRIX.  
 COLUMN 1 ROWS 1 THRU 3 -----  
 2.70321E+01 -1.92164E+02 2.10153E+02  
 THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3  
 THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.  
 \*\*\* SYSTEM WARNING MESSAGE 2172, ROW AND COLUMN PARTITIONING VECTORS DO NOT HAVE IDENTICAL ORDERING OF ZERO AND NON-ZERO ELEMENTS, AND SYM FLAG INDICATES THAT A SYMMETRIC PARTITION OR MERGE IS TO BE PERFORMED.  
 \*\*\* SYSTEM WARNING MESSAGE 2172, ROW AND COLUMN PARTITIONING VECTORS DO NOT HAVE IDENTICAL ORDERING OF ZERO AND NON-ZERO ELEMENTS, AND SYM FLAG INDICATES THAT A SYMMETRIC PARTITION OR MERGE IS TO BE PERFORMED.

MATRIX Y (GINO NAME 101 ) IS A REAL 3 COLUMN X 3 ROW SYMETRIC MATRIX.  
 COLUMN 1 ROWS 1 THRU 3 -----  
 1.00000E+00 5.00000E-01 3.33330E-01  
 COLUMN 2 ROWS 1 THRU 3 -----  
 5.00000E-01 3.33330E-01 2.50000E-01  
 COLUMN 3 ROWS 1 THRU 3 -----  
 3.33330E-01 2.50000E-01 2.00000E-01  
 THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3  
 THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.

MATRIX CP (GINO NAME 102 ) IS A REAL 1 COLUMN X 3 ROW RECTANG MATRIX.  
 COLUMN 1 ROWS 1 THRU 2 -----  
 1.00000E+00 1.00000E+00  
 THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 2  
 THE DENSITY OF THIS MATRIX IS 66.66 PERCENT.

APPENDIX

JANUARY 24, 1977 NASTRAN 1/ 1/77 PAGE 12

MATRIX RP (GIND NAME 103 ) IS A REAL 1 COLUMN X 3 ROW RECTANG MATRIX.  
 COLUMN 1 ROWS 2 THRU 3 -----  
 1.00000E+00 1.00000E+00  
 THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 2  
 THE DENSITY OF THIS MATRIX IS 66.66 PERCENT.

JANUARY 24, 1977 NASTRAN 1/ 1/77 PAGE 13

MATRIX A11 (GIND NAME 101 ) IS A REAL 1 COLUMN X 1 ROW SYMETRIC MATRIX.  
 COLUMN 1 ROWS 1 THRU 1 -----  
 3.33330E-01  
 THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 1  
 THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.

JANUARY 24, 1977 NASTRAN 1/ 1/77 PAGE 14

MATRIX A21 (GIND NAME 102 ) IS A REAL 1 COLUMN X 2 ROW RECTANG MATRIX.  
 COLUMN 1 ROWS 1 THRU 2 -----  
 2.50000E-01 2.00000E-01  
 THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 2  
 THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.

JANUARY 24, 1977 NASTRAN 1/ 1/77 PAGE 15

MATRIX A12 (GIND NAME 103 ) IS A REAL 2 COLUMN X 1 ROW RECTANG MATRIX.  
 COLUMN 1 ROWS 1 THRU 1 -----  
 1.00000E+00  
 COLUMN 2 ROWS 1 THRU 1 -----  
 5.00000E-01  
 THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 1  
 THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.

JANUARY 24, 1977 NASTRAN 1/ 1/77 PAGE 16

MATRIX A22 (GIND NAME 104 ) IS A REAL 2 COLUMN X 2 ROW SYMETRIC MATRIX.  
 COLUMN 1 ROWS 1 THRU 2 -----  
 5.00000E-01 3.33330E-01  
 COLUMN 2 ROWS 1 THRU 2 -----  
 3.33330E-01 2.50000E-01  
 THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 2  
 THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.

METHOD 1 NT,NRR PASSES = 1, EST. TIME = 00.0

\*\*\*USER INFORMATION MESSAGE 3023--PARAMETERS FOR SYMMETRIC DECOMPOSITION OF DATA BLOCK A ( N = 3 )  
 TIME ESTIMATE= 1 C AVG = 2 PC AVG = 0 SPILL GROUPS = 0 S AVG = 1  
 ADDITIONAL CORE= -15660 C MAX = 3 PCMAX = 0 PC GROUPS = 0 PREFACE LOOPS = 1

# APPENDIX

```

MATRIX X      (GINO NAME 101 ) IS A REAL      1 COLUMN X      3 ROW RECTANG MATRIX.
COLUMN 1      ROWS 1 THRU 3 -----
2.70321E+01 -1.92164E+02 2.10153E+02
THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3
THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.
    
```

```

MATRIX W      (GINO NAME 102 ) IS A REAL      1 COLUMN X      3 ROW RECTANG MATRIX.
COLUMN 1      ROWS 1 THRU 3 -----
2.70321E+01 -1.92164E+02 2.10153E+02
THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3
THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.
    
```

```

MATRIX Z      (GINO NAME 103 ) IS A REAL      1 COLUMN X      3 ROW RECTANG MATRIX.
COLUMN 1      ROWS 1 THRU 3 -----
2.99999E+00 1.91666E+00 1.43333E+00
THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3
THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.
VALUE OF ELEMENT T ( 1, 1) = ( 1.0000000E+00 , 0.E+00 )
    
```

```

MATRIX V      (GINO NAME 101 ) IS A REAL      3 COLUMN X      3 ROW SYMETRIC MATRIX.
COLUMN 1      ROWS 1 THRU 3 -----
1.00000E+00 5.00000E-01 3.33330E-01
COLUMN 2      ROWS 1 THRU 3 -----
5.00000E-01 3.33330E-01 2.50000E-01
COLUMN 3      ROWS 1 THRU 3 -----
3.33330E-01 2.50000E-01 2.00000E-01
THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3
THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.
    
```

# APPENDIX

```

MATRIX T      (GIND NAME 102 ) IS A REAL      1 COLUMN X      3 ROW RECTANG MATRIX.
COLUMN 1      ROWS 1 THRU 3 -----
1.00000E+00 1.00000E+00 1.00000E+00
THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3
THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.
*** USER INFORMATION MESSAGE 4114.
DATA BLOCK A WRITTEN ON FORTRAN UNIT 11, TRL =
101          3          3          6          1          3          10000          1
*** USER INFORMATION MESSAGE 4114.
DATA BLOCK B WRITTEN ON FORTRAN UNIT 11, TPL =
102          1          3          2          1          3          10000          1
*** USER INFORMATION MESSAGE 4105. DATA BLOCK C RETRIEVED FROM FORTRAN TAPE 11
NAME OF DATA BLOCK WHEN PLACED ON FORTRAN TAPE WAS A
*** USER INFORMATION MESSAGE 4105. DATA BLOCK D RETRIEVED FROM FORTRAN TAPE 11
NAME OF DATA BLOCK WHEN PLACED ON FORTRAN TAPE WAS B
    
```

```

MATRIX C      (GIND NAME 101 ) IS A REAL      3 COLUMN X      3 ROW SYMETRIC MATRIX.
COLUMN 1      ROWS 1 THRU 3 -----
1.00000E+00 5.00000E-01 3.33330E-01
COLUMN 2      ROWS 1 THRU 3 -----
5.00000E-01 3.33330E-01 2.50000E-01
COLUMN 3      ROWS 1 THRU 3 -----
3.33330E-01 2.50000E-01 2.00000E-01
THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3
THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.
    
```

```

MATRIX D      (GIND NAME 102 ) IS A REAL      1 COLUMN X      3 ROW RECTANG MATRIX.
COLUMN 1      ROWS 1 THRU 3 -----
1.00000E+00 2.00000E+00 3.00000E+00
THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3
THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.
    
```

```

MATRIX E      (GIND NAME 101 ) IS A REAL      3 COLUMN X      3 ROW SYNETPIC MATRIX.
COLUMN 1      ROWS 1 THRU 3 -----
1.00000E+00 5.00000E-01 3.33330E-01
COLUMN 2      ROWS 1 THRU 3 -----
5.00000E-01 3.33330E-01 2.50000E-01
COLUMN 3      ROWS 1 THRU 3 -----
3.33330E-01 2.50000E-01 2.00000E-01
THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3
THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.
    
```

# APPENDIX

MATRIX A (GINO NAME 102 ) IS A REAL 3 COLUMN X 3 ROW SYMETRIC MATRIX.

COLUMN	1	ROWS	1 THRU	3	-----
	1.00000E+00	5.00000E-01	3.33330E-01		
COLUMN	2	ROWS	1 THRU	3	-----
	5.00000E-01	3.33330E-01	2.50000E-01		
COLUMN	3	ROWS	1 THRU	3	-----
	3.33330E-01	2.50000E-01	2.00000E-01		

THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3

THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.

MATRIX B (GINO NAME 103 ) IS A REAL 1 COLUMN X 3 ROW RECTANG MATRIX.

COLUMN	1	ROWS	1 THRU	3	-----
	1.00000E+00	2.00000E+00	3.00000E+00		

THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD = 3

THE DENSITY OF THIS MATRIX IS 100.00 PERCENT.

## CONTENTS OF PARAMETER TABLE

MINDIAG	0
DET	0
POWER	0
SING	0
ISH	-1
X2	0
X1	2.000000E+00

## CONTENTS OF PARAMETER TABLE

MINDIAG	0
DET	0
POWER	0
SING	0
ISH	-1
X2	0
X1	3.000000E+00



APPENDIX

JANUARY 24, 1977 NASTRAN 1/ 1/77 PAGE 29

C O N T E N T S O F P A R A M E T E R T A B L E

MINDIAG	0
DET	0
POWER	0
SING	0
ISW	-1
X2	0
X1	4.000000E+00

JANUARY 24, 1977 NASTRAN 1/ 1/77 PAGE 30

C O N T E N T S O F P A R A M E T E R T A B L E

MINDIAG	0
DET	0
POWER	0
SING	0
ISW	-1
X2	0
X1	5.000000E+00

## REFERENCES

1. The NASTRAN Theoretical Manual (Level 16.0). NASA SP-221(03), 1976.
2. The NASTRAN User's Manual (Level 16.0). NASA SP-222(03), 1976.
3. The NASTRAN Programmer's Manual (Level 16.0). NASA SP-223(03), 1976.
4. The NASTRAN Demonstration Problem Manual (Level 16.0). NASA SP-224(03), 1976.

☆U.S. GOVERNMENT PRINTING OFFICE: 1978-735-078/25

1. Report No. NASA RP-1019		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle  GUIDE TO A CONDENSED FORM OF NASTRAN®				5. Report Date September 1978	
				6. Performing Organization Code	
7. Author(s) James L. Rogers, Jr.				8. Performing Organization Report No. L-11900	
9. Performing Organization Name and Address  NASA Langley Research Center Hampton, VA 23665				10. Work Unit No. 506-17-21-01	
				11. Contract or Grant No.	
12. Sponsoring Agency Name and Address  National Aeronautics and Space Administration Washington, DC 20546				13. Type of Report and Period Covered Reference Publication	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract  A condensed form of NASTRAN® Level 16 is described. Included are descriptions of the input cards, the programming language of the direct matrix abstraction program, the plotting, the problem definition, the modules' diagnostic messages, and sample problems that relate to the capabilities retained in the condensed form. This guide can serve as a handbook for instructional courses in the use of NASTRAN or for users who only need the capability provided by the condensed form.					
17. Key Words (Suggested by Author(s))  NASTRAN® Finite element Computer program			18. Distribution Statement  Unclassified - Unlimited  Subject Category 61		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 296	22. Price* \$11.00

National Aeronautics and  
Space Administration

Washington, D.C.  
20546

Official Business  
Penalty for Private Use, \$300

SPECIAL FOURTH CLASS MAIL  
BOOK

Postage and Fees Paid  
National Aeronautics and  
Space Administration  
NASA-451



6 1 1U,G,SPGEN,081178 S00903DS 740731  
DEPT OF THE AIR FORCE  
AF WEAPONS LABORATORY  
ATTN: TECHNICAL LIBRARY (SUL)  
KIRTLAND AFB NM 87117

**NASA**

POSTMASTER: If Undeliverable (Section 158  
Postal Manual) Do Not Return

---