NASA CR-
141697

COMPUTER PROGRAM DOCUMENTATION

DOCUMENTATION OF STRUCTURES BRANCH PROGRAMS AND PROGRAM UPDATES

(NASA-CR-141697) DOCUMENTATION OF STRUCTURES BRANCH PROGRAMS AND PROGRAM UPDATES. PROJECT 3200 (Lockheed Electronics Co.) 163 p HC $6.25 CSCL 09B

N75-18921

Unclas
G3/61 13529

Prepared by

Donald G. Probe

Applied Mechanics Department
Lockheed Electronics Company, Inc.
Aerospace Systems Division
Houston, Texas

Prepared Under Contract NAS 9-12200

For

STRUCTURES AND MECHANICS DIVISION

January 1975

National Aeronautics and Space Administration
LYNDON B. JOHNSON SPACE CENTER
Houston, Texas

LEC-5442

JSC-09404

Applied Mechanics
Department 641-11
CPD 411

COMPUTER PROGRAM DOCUMENTATION

DOCUMENTATION OF STRUCTURES BRANCH PROGRAMS AND PROGRAM UPDATES

Project 3200

Prepared by

Donald G. Probe

Applied Mechanics Department
Lockheed Electronics Company, Inc.
Aerospace Systems Division
Houston, Texas

Prepared Under Contract NAS 9-12200

For

STRUCTURES AND MECHANICS DIVISION

National Aeronautics and Space Administration
Johnson Space Center
Houston, Texas

January 1975

LEC-5442

JSC-09404

Applied Mechanics
Department 641-11
CPD 411

COMPUTER PROGRAM DOCUMENTATION
DOCUMENTATION OF STRUCTURES BRANCH
PROGRAMS AND PROGRAM UPDATES

PREPARED BY

D. G. Probe, Supervisor
Structures Technology Section
Lockheed Electronics Company, Inc.

APPROVED BY

LEC

R. T. Theobald, Manager
Applied Mechanics Department

NASA

R. L. Nieder
ISAS, Category 1
Documentation Manager

R. D. Schwartz
Task Monitor
ISAS Category 1 Development

Prepared by

Lockheed Electronics Company, Inc.

For

Structures and Mechanics Division

Aerospace Systems Division
Lyndon B. Johnson Space Center
Houston, Texas

January 1975

LEC-5442

CONTENTS

	Page
1.0 INTRODUCTION	1-1
2.0 DESCRIPTION OF PROGRAM DOCUMENTATION	2-1
2-Abstract	2-3
2-1.0 Introduction	2-4
2-2.0 Program Description	2-5
2-2.1 General Description	2-5
2-2.2 Technical Description	2-5
2-2.2.1 Analytical and Problem Formulation	2-6
2-2.2.2 Solution Considerations	2-6
2-2.N Additional Subsections	2-6
2-3.0 Program Usage	2-7
2-3.1 Program Organization and Operating Characteristics	2-7
2-3.2 Input Description	2-7
2-3.3 Output Description	2-7
2-3.4 Sample Problems	2-8
2-3.4.1 Problem Description . .	2-8
2-3.4.2 Input Deck	2-8
2-3.4.3 Sample Problem Output .	2-8
2-3.5 Discussion of Program Characteristics	2-9
2-4.0 Subroutine Documentation	2-10
2-4.1 Main Program Description	2-10
2-4.2 Common Block Information	2-10
2-4.3 Namelist Information	2-11
2-4.4 Parameter Information	2-11
2-4.5 Subroutine Description	2-11

CONTENTS (continued)

Page

2-4.5.1 Subroutine ABCDEF (first subroutine). . . . 2-11

2-4.5.2 Additional Subroutines . 2-12

2-4.5.N Additional Subroutines . 2-12

3.0 UTILIZATION OF DOCUMENTATION PROCEDURES 3-1

APPENDICES

A - SCIENTIFIC PROGRAM DOCUMENTATION CHECKLIST A-1

B - SAMPLE OF SUBROUTINE DOCUMENTATION B-1

C - A SAMPLE DOCUMENT WRITTEN TO FOLLOW THE SUGGESTED FORMAT C-1

1.0 INTRODUCTION

There is a large amount of programming being performed by the LEC, Applied Mechanics Department, Structures Technology Section for the purpose of supporting the original development and update programming of applications programs for the Integrated Structural Analysis System (ISAS). This discussion and outline are presented in an attempt to layout a standard document format which will be rigidly held to in the preparation of program documents.

The fact that documentation is required for all programming work performed by LEC for the NASA/JSC, Structures and Mechanics Division, Structures Branch has previously been agreed to by both parties. The documentation thus far has been satisfactory for the most part but there has been a persistent inconstancy in the document format and content. It seems that the largest difficulty is the programmers requirement that both a program document and a users guide must be combined as one document.

A review of the time being spent on documentation versus time estimates on task agreements indicates that documentation effort is being underestimated. The additional information, formatting, gathering, and preparing considered herein may cause future documentation estimates to double past estimates but this will still be a small portion of the total task effort.

The document suggested to alleviate the existing difficulties is described within this guide. It is designed so both programmers and program users who desire or are required to work with the program sometime in the future will have sufficient information with which to use or alter the program. The intended document format is outlined and described in

Section 2 of this guide.

Documentation which involves changes, additions, and I/O capability revisions to existing programs will follow the format of Section 2 as closely as practical. The documentation in this case will not normally involve the entire subject program, only those portions which have been added or modified.

There are sufficient appendices included in this guide to give the programmer a good idea of what is intended by the verbal descriptions in Section 2. Appendix A consists of a checklist which should be reviewed each time a programming effort is documented. Each item which appears on the checklist and pertains to the programming performed should be included in the planned document.

2.0 DESCRIPTION OF PROGRAM DOCUMENTATION

Each subsection of Section 2 is a description of an identifiable portion of the intended program document. The outline of the new format is shown in Figure 2-1.

There will be no discussion of the title page, approval page and table of contents within this guide, except to say that previous formats for technical memos and program documentation are acceptable as they have been in the past. The specific format here will vary with the importance of the subject document and the formality with which it is felt that it should be documented.

The numbering of subsections within Section 2 will be like so:

2-1.0
2-2.0
2-2.1
2-2.1.2
⋮
etc.,

so as to be able to identify these subsections from primary subsections of this document.

TITLE PAGE

APPROVAL PAGE

TABLE OF CONTENTS

ABSTRACT

1.0 INTRODUCTION

2.0 PROGRAM DESCRIPTION

- 2.1 General Description
- 2.2 Technical Description
  - 2.2.1 Analytical & Problem Formulation
  - 2.2.2 Solution Considerations
- 2.N As Required

3.0 PROGRAM USAGE

- 3.1 Program Organization
- 3.2 Input Description
- 3.3 Output Description
- 3.4 Sample Problems
  - 3.4.1 Problem Description
  - 3.4.2 Input Deck
  - 3.4.3 Sample Problem Output
- 3.5 Discussion of Program Characteristics

4.0 MAIN PROGRAM AND SUBROUTINE DOCUMENTATION

- 4.1 Main Program Documentation
- 4.2 Common Block Information
- 4.3 Namelist Description
- 4.4 Subroutine Documentation

5.0 REFERENCES

APPENDIX

FIGURE 2-1 OUTLINE FOR PROGRAM DOCUMENTATION

2 - ABSTRACT

The abstract is a concise description of the capabilities of the program. Whenever possible it will be written on JSC form 143, entitled, "JSC Computer Program Abstract". The abstract will describe program input requirements, the program's operational characteristics, and the program output in general terms. The programming language and the required computer facility description for using the program will be included.

A program number will be obtained from the ADP Computer Program Sharing Library when the subject of the document is a newly developed program.

2-1.0 INTRODUCTION

The introduction contains the purpose of the program including background data leading to the request for the program development. The relationship of the subject program with other programs and/or systems is discussed in this section. A flow chart showing interrelationships may be placed in this section if it adds to the clarity of the explanation. Also included in this section will be a description of the programs scope or problem solving capability.

Other items which may appear in this section:

- The program name and acronym if any.
- Broad description of planned input sources.
- General type and size of problems which the program will solve.
- Broad output description and where the output is intended to be utilized.

2-2.0 PROGRAM DESCRIPTION

This section is intended primarily for the analyst who is interested in a description of the technical details and solution procedures used in the development of the program. The engineering or scientific problem formulation contained in the program will be presented in this section. Also the methods for achieving a numerical solution to the formulated problem will be described here. Broad statements leading into the following more detailed subsections will be presented in Section 2.

2-2.1 General Description

Begin to describe in detail all the analytical capabilities which have been designed into the program. A physical description of all the required input information will be presented in this subsection along with a similar description of the program's output. An explanation of why this input and output is required, optional program capabilities, and I/O will be outlined in this section.

2-2.2 Technical Description

This subsection contains the technical solution flow description and organization of the program. Flow charts detailing the path through the analytical solution may be presented if they are helpful. The method of applying numerical procedures for solving the analytical problem are described. It may be of some value and clear-up the presentation for large programs if the technical description, Section 2.2, is split into two headings: the first, Analytical and Problem Formulation; the second, Solution Considerations.

2-2.2.1 Analytical and Problem Formulation

The actual engineering or scientific problem to be solved by the program in equation form should be detailed in this subsection. This section could start with a nomenclature page to put symbol explanations in close proximity to the symbol equations. The program variable name could also be related to the symbols and nomenclature if it would clarify the document. Several subsections might be used to describe various analytical methods which might be used within one program.

2-2.2.2 Solution Considerations

Logical construction of the program design to achieve problem solution will be spelled out in this section. All numerical techniques used for equation solution will be described in detail within this subsection. Additional subsections may be used in large size programs for clarification. If additional function flow charts can further clarify the explanation they will be included here.

2-2.N Additional Subsections

It may be appropriate to describe the functions of the mathematical subroutines briefly if analytical flow charts are presented within section 2.0. It may depend on the program individuality whether subroutine functions will better be described here or within the next section, 3.0, Usage. Other items which could appear in this section

- Information to explain details and relate program sections
- Definition of Major Internal Variables

2-3.0 PROGRAM USAGE

This section is intended to be the program users manual. It must contain sufficient information to enable a new program user to successfully utilize all the capabilities of the program. Input, output, special characteristics and sample problems must be thoroughly described and discussed.

2-3.1 Program Organization and Operating Characteristics

An overall program flow chart will be presented in this subsection. This overall flow chart can be used as reference throughout this section.

2-3.2 Input Description

A detailed description of the input required for all options of the program will be placed within this subsection. Other data to appear here will include:

- A deck setup description
- Input data detailed formats for card, tape and drum input - also terminal
- Description of input data and units required
- Program symbolic names and mathematical names
- Reference to sample problem input data

2-3.3 Output Description

A detailed description of all output options available to the user. Data to be included here will include:

- Diagnostic descriptions and solutions

-9-

- Detailed output descriptions of content and formats
- References to sample problems
- Where and how output is used - hard copy, tapes, plots
- Distinguish between calculated output and output which is echoed input

2-3.4 Sample Problems

All the information necessary to reproduce the sample problem should be contained and explained in this subsection. Program features and options should be pointed out. Comparison of results with an independent solution should be made if possible.

2-3.4.1 Problem Description

The physical problem being solved must be completely described here. Physical and geometrical properties required for the solution must be supplied or the method of obtaining them defined. Drawings or sketches clarifying the problem are to be included.

2-3.4.2 Input Deck

The sample problem input deck is to be shown and explained by relating input parameters to physical quantities. Input file formats are to be shown and explained. Input tape units are to be identified along with input tape format descriptions.

2-3.4.3 Sample Problem Output

A description, explanation, and discussion of sample problem output using hardcopy listings as a reference figure.

2-3.5 Discussion of Program Characteristics

A discussion with reference to the sample problem of such program characteristics as run time, number of pages output, special features and options, and various modes of output.

2-4.0 SUBROUTINE DOCUMENTATION

This section of the program document is intended to provide significant information related to the programming aspects of the program. Such data as the language it is written in, the computer and system it was written for, overlay information, and the required core storage for the program should be written into this section. Other pertinent information regarding future improvements, existing problems, and special program features should be spelled out here.

A light introduction to the documentation of program routines could also appear in this section.

Refer to Appendix B and C for examples of subroutine documentation.

2-4.1 Main Program Description

A description of the purpose, function, and organization of the main routine of the program is presented in this section. An overall program flow chart should be presented unless the program is a minor one. An overlay listing and a listing of the main program code should appear in this part.

2-4.2 Common Block Information

Programs having several named common blocks should use this section to describe them. A listing of the variables contained in each block should be shown. Usually these variables will be defined within the section describing each subroutine so they probably will not have to be defined here. The use of Fortran Procedures to incorporate common blocks within various subroutines should be described if used. A common block versus subroutine matrix showing which common blocks are contained in each subroutine should be placed in this section.

2-4.3 Namelist Information

Describe namelists used in the program. Associate the namelist names with the variables within each.

2-4.4 Parameter Information

In a case where extensive use is made of parameter constants a section such as this can be used to identify and describe the parameters and their use.

2-4.5 Subroutine Description

A general description of each subroutine used within the program and its use should be written in this section.

2-4.5.1 Subroutine ABCDEF (First subroutine)

For each subroutine of significance the following data will be included:

- Subroutine function
- Required subroutine storage
- List of important parameters
- List of important input variables
- List of important output variables
- Description of tape and other mass storage input
- Description of tape and other mass storage output
- List of error messages and method of error correction

- Printed output description
- Library routines required
- Flow chart for large subroutines
- Listing

2-4.5.2 Additional Subroutine

2-4.5.N Additional Subroutine

-14-

2-5.0 REFERENCES

As Required

2 - APPENDICES

As Required

3.0 UTILIZATION OF DOCUMENTATION PROCEDURES

The procedures described in this documentation guide will result in large documents. The documents will contain everything useful that there is to know about the subject program except for extensive user experience descriptions. The information within the documents, written to the format described herein, will have analytical data presented from an analysts point of view within Section 2., user information presented from a users point of view within Section 3., and programmers information presented from a programmers point of view presented within Section 4.

This guide will be distributed to all programmers performing category 1 program development. They will be instructed to use this guide for reference when documenting new programs and making changes and additions to existing programs. Appendix A, a checklist to be used when writing a program document, will be reviewed when each program document is written.

It is intended that a continuing review of documentation will be exercised so as to insure that the Structures Branch of the Structures and Mechanics Division receives the most efficient program documentation that can be written. Continuing feedback from technical monitors, program and document users regarding documentation quantity and quality is expected, helpful, and appreciated.

APPENDIX A

Scientific Program Documentation Checklist

SCIENTIFIC PROGRAM DOCUMENTATION

CHECKLIST

ABSTRACT

__ JSC Form 143

1. INTRODUCTION

__ Program Purpose (general)
__ Background Information
__ Source of Input
__ Relationships to other progra
__ Program Scope
__ Output Utilization

2. PROGRAM DESCRIPTION (Equations and Method of Solution)

- General Description
  - __ Program purpose (specific) (physical significance)
  - __ Program capabilities, options
  - __ Computational philosophy/logical development
  - __ Input/output parameters
  - __ Input/output media utilized
- Technical Description
  - ●● Analysis
    - __ Problem formulation
    - __ Equations (math model)
    - __ Math symbol definitions
  - ●● Method of solution
    - __ System design
    - __ Numerical techniques
    - __ General flow chart

3. USAGE (Program users manual)

__ Overall Functional Flow Chart

-18-

- Input Description
  - __ Data specifications (type, format, units, options)
  - __ Card specifications (card columns vs data identificatio
  - __ Tape specifications (record, word vs data identificatio

- Program Run Preparations
  - __ Deck set-up
  - __ Special control cards
  - __ Special I/O devices
  - __ Non-standard routines
  - __ Overlay structure

- Output Description
  - __ Identification-printer output labels and headings
  - __ Identifications-non-printer (plotter, punched card, etc.) labels and headings
  - __ Variable descriptions (format, units of measure)
  - __ Tape output (record, file, word definitions)
  - __ Programmed diagnostics (message, source, and action requir
  - __ References to sample problems

- Execution Characteristics

  - •• Restrictions
    - __ Analytic
    - __ Hardware
    - __ Storage
    - __ Programmed diagnostics referenced

  - •• Running Time/Lines of Output
    - __ Method of estimation
    - __ Maximum estimate

  - •• Accuracy/Validity
    - __ Double precision arithmetic operations
    - __ Loss of numeric significance
    - __ Relation of output significance to input
    - __ Validation or verification of program (calculator or comparison with other programs)

- Sample Problem
  - __ Problem Description
  - __ Input deck, tapes, and files
  - __ Output description and explanation
  - __ Plots

-19-

4. SUBROUTINE DOCUMENTATION (Programmer Document)

- Main Program Description
  - __ Overall flow chart
  - __ Listing
  - __ Map
  - __ Input
  - __ Output
- Common Block Information
  - __ Cross reference matrix
- Namelist Information
- Parameter Information
- Subroutine Description
  - __ Individual routine functions
  - •• Individual Subroutine Description
    - __ Function
    - __ Storage
    - __ Parameters
    - __ Input variables
    - __ Output variables
    - __ Tape and mass storage input
    - __ Tape and mass storage output
    - __ Error messages and correction
    - __ Normal output
    - __ Library routines
    - __ Flow chart
    - __ Listing

5. REFERENCES

- __ Technical references
- __ Related program(s), routine(s),task descriptions

APPENDICES

- Tables
  - __ Tables in appendix (too large for text)
- Figures
  - __ Figures in appendix (too large for text)
- Listings
  - __ Listings in appendix (too large for text)

APPENDIX B

Sample of Subroutine Documentation

(3 subroutines included)

4.4 Subroutine Description

The functions of the subroutines which have been added to the FRISBE Program and the added capabilities of altered subroutines are described below.

ROUTINE	FUNCTION
DREAD	Additional namelist capability has been added to this subroutine so controlling data for reading flight condition and force coefficient files can be read into the program.
FCREAD	A routine which reads a flight conditions file from tape. This routine will not necessarily read the ISAS Flight Condition File resident on drum storage. A start time and an end time are obtained from namelist input in DREAD. All flight conditions between the start and end time are read and stored in core (COMMON Block AERO).
FILE	This routine replaces the EXEC-2 library routine also called FILE. There is no similar routine in the EXEC-8 library so this routine was written in Fortran to perform the previous function of the EXEC-2 FILE routine. The routine skips a specified number of end-of-file marks before beginning a read or write.
FOCORD	A routine which reads force coefficients from an ISAS formatted Force Coefficient Data File (on drum storage) for the flight conditions which are indicated from namelist input or from the Flight Conditions File.

23

ROUTINE	FUNCTION
INTEG	The output of the ISAS Flight Conditions Tape File (FRISBE output file) was altered slightly to make the format compatible with the other "Batch Flight Condition Files".
INTERP	This subroutine has been altered to include the capability of interpolating two force data files to construct a combined applied force file for dynamic response analysis.
TREAD	The previously existing subroutine which is used primarily for reading tape data input has been altered to control the reading of the flight conditions data from tape and the force coefficient data from drum file storage. The subroutine also controls the force interpolation used in the construction of an external force tape.

4.4.1 Subroutine DREAD

The purpose of subroutine DREAD is to read namelist data in namelist $IPGFP. The added variables to be read are control parameters for reading ISAS files and accuracy requirements for selection of force coefficient sets from flight condition specifications. An output control parameter may also be read through $IPGFP

STORAGE:

IBANK	1605_8
DBANK	14000_8

PARAMETERS:	SIZE	DESCRIPTION
NIMAX	25	Number of interface loads - max.
NFMAX	200	Number of external forces - max.
NLMAX	75	Number of nodal point accelerations - max.
NTMAX	50	Number of time points - max.
NNMAX	150	Number of nodes - max.
NDMAX	400	Number of degrees-of-freedom - max.
NMMAX	50	Number of modes - max.

INPUT VARIABLES:

MACH	R	Mach number array
ALPHA	R	Angle-of-attack array
BETA	R	Yaw angle array
DELTAE	R	Elevon angle array
DELTAR	R	Rudder angle array
FCFILE	L	If true read flight conditions tape
FOCOFL	L	If true read force coefficient file
CNFGCK	A	Configuration ID word
TIME	R	Flight condition time array
QPSI	R	Dynamic pressure array
MACDIF	R	Accuracy requirement for MACH array
ALFDIF	R	Accuracy requirement for ALPHA array
BETADF	R	Accuracy requirement for BETA array
DELEDF	R	Accuracy requirement for DELTAE array
DELRDF	R	Accuracy requirement for DELTAR array
OPFCHK	I	Print control

OUTPUT VARIABLES:

MACH		↑
ALPHA		
BETA		
DELTAE		Refer to input variables
DELTAR		
FCFILE		
FOCOFL		↓
STARTM	R	Starting time for the run
ENDTIM	R	End time for the run
TIME		↑
QPSI		
CNFGCK		
MACDIF		
ALFDIF		Refer to input variables

-25-

BETADF
DELEDF
DELRDF
OPFCHK

Refer to input variables ↓

PRINTED OUTPUT:

The printed output from subroutine DREAD includes the modal tape title and the case title followed by namelist OUTPT. Namelist OUTPT is a previously existing namelist to which has been added the variables MACH, ALPHA, BETA, DELTAE, DELTAR, FCFILE, FOCOFL, STARTM, ENDTIM, and OPFCHK.

LISTING:

A listing of subroutine DREAD is shown on the following eight pages.

-26-

```
******  DREAD  ******

@FOR,S P.DREAD,DREAD
FOR SEIX-11/13/74-01:07:30 (17,)

   SUBROUTINE DREAD     ENTRY POINT 001564

   STORAGE USED: CODE(1) 001605; DATA(0) 014000; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    ALOGIC 000002
    0004    CASDAT 000014
    0005    CDATA  000012
    0006    CLOGIC 000004
    0007    OLOGIC 000005
    0010    SYMLOG 000005
    0011    PLOGIC 000016
    0012    SCRDRM 000006
    0013    IPTAPE 000005
    0014    OPTAPE 000004
    0015    TLOGIC 000005
    0016    OUTSAS 000002
    0017    AERO   000574

   EXTERNAL REFERENCES (BLOCK, NAME)

    0020    FILE
    0021    NRNLS
    0022    NREWS
    0023    NRBUS
    0024    NI02S
    0025    NPRTS
    0026    NI01S
    0027    NI03S
    0030    NWNLS
    0031    NWBUS
    0032    NWEFS
    0033    NERR3S

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)
```

0001	000147 100L	0001	001157 1000G	0001	001160 1002G	0001	001171 1007G	0001	001202 1015G
0001	001202 1017G	0001	001207 1024G	0001	001207 1026G	0001	001222 1036G	0001	001222 1040G
0001	001227 1045G	0001	001227 1047G	0001	001234 1054G	0001	001234 1056G	0001	001245 1065G
0001	001252 1071G	0001	001257 1075G	0001	000163 110L	0001	001275 1107G	0001	001302 1113G
0001	001347 1141G	0001	001347 1143G	0001	001354 1150G	0001	001354 1152G	0001	001403 1166G
0001	001410 1172G	0001	001415 1176G	0001	000214 120L	0001	001472 1232G	0001	001511 1242G
0001	001531 1253G	0001	001532 1255G	0001	001536 1261G	0001	000231 130L	0001	000244 140L
0001	000256 150L	0001	000310 160L	0000	013610 170F	0000	013622 180F	0001	000327 190L
0000	013631 200F	0001	000046 207G	0000	013642 210F	0001	000053 216G	0001	000515 220L
0001	000057 224G	0001	000520 230L	0001	000067 233G	0001	000070 236G	0001	000532 240L
0001	000074 245G	0001	000104 254G	0001	000557 260L	0001	000115 267G	0001	000565 270L
0001	000577 280L	0001	000604 290L	0001	000124 300G	0001	000617 300L	0001	000131 307G
0000	013652 310F	0001	000132 312G	0000	013671 320F	0000	013677 340F	0000	013721 350F

22

```
0001     001060 370L    0001     001515 375L    0001     001427 380L    0001     001543 385L    0001     000343 420G
0001     000400 440G    0001     000412 445G    0001     000417 451G    0001     000420 453G    0001     000431 460G
0001     000506 503G    0001     000545 527G    0001     000660 572G    0001     000667 600G    0001     000711 614G
0001     000736 631G    0001     000743 635G    0001     000750 641G    0001     000761 647G    0001     000771 656G
0001     001124 753G    0001     001131 757G    0001     001136 763G    0001     001152 774G    0000 R 000751 AFTXYZ
0017 R 000570 ALFDIF    0000 L 000001 ALLMOD    0017 R 000062 ALPHA     0010 L 000003 ANTI      0017 R 000144 BETA
0017 R 000571 BETADF    0000 R 012514 BGNPRT    0000 R 010573 BGNTIM    0010 L 000004 CGCALC    0003 L 000001 CKOUT
0017 I 000374 CNFGCK    0006 L 000002 COMPIC    0004 R 000000 CTITLE    0000 R 006145 DAMP      0017 R 000572 DELEDF
0017 R 000573 DELRDF    0017 R 000226 DELTAE    0017 R 000310 DELTAR    0000 R 010606 DELTIM    0000 R 006311 DISPL
0000 R 010620 DMASS     0000 R 012536 DMP       0000 R 012515 ENDPRT    0017 R 000421 ENDTIM    0000 L 000002 ENGID
0003 L 000000 ERROR     0000 L 000003 EXTFOR    0017 L 000372 FCFILE    0017 L 000373 FOCOFL    0000 I 000005 FORMID
0015 L 000000 FORTAP    0000 R 000740 FORXYZ    0015 L 000004 FPLOT     0000 R 007275 FTILE     0000 R 001075 F1
0000 R 001777 F10       0000 R 002061 F11       0000 R 002143 F12       0000 R 002225 F13       0000 R 002307 F14
0000 R 002371 F15       0000 R 002453 F16       0000 R 002535 F17       0000 R 002617 F18       0000 R 002701 F19
0000 R 001157 F2        0000 R 002763 F20       0000 R 003045 F21       0000 R 003127 F22       0000 R 003211 F23
0000 R 003273 F24       0000 R 003355 F25       0000 R 003437 F26       0000 R 003521 F27       0000 R 003603 F28
0000 R 003665 F29       0000 R 001241 F3        0000 R 003747 F30       0000 R 004031 F31       0000 R 004113 F32
0000 R 004175 F33       0000 R 004257 F34       0000 R 004341 F35       0000 R 004423 F36       0000 R 004505 F37
0000 R 004567 F38       0000 R 004651 F39       0000 R 001323 F4        0000 R 004733 F40       0000 R 005015 F41
0000 R 005077 F42       0000 R 005161 F43       0000 R 005243 F44       0000 R 005325 F45       0000 R 005407 F46
0000 R 005471 F47       0000 R 005553 F48       0000 R 005635 F49       0000 R 001405 F5        0000 R 005717 F50
0000 R 001467 F6        0000 R 001551 F7        0000 R 001633 F8        0000 R 001715 F9        0000 R 012535 GMASS
0006 L 000000 HOLDIT    0000 I 012529 I         0000 I 012537 IC        0013 I 000000 ICARDS    0000 I 012540 IC2
0000 I 010460 IDACC     0012   000003 IDRUM     0000 I 011440 IDXYZ     0013   000004 IFCFIL    0017   000005 IFUSCR
0013   000003 IFCTAP    0013   000002 IFIN      0012   000001 IFORC     0014   000002 IFPLOT    0014 I 000003 ILINKS
0000   013745 INJPS     0013 I 000001 INPUT     0014   000000 IOPDR     0000   012560 IPSFP     0012 I 000002 IPHIS
0014   000001 IRPT      0016 L 000000 ISAS      0012   000004 ISCRT     0016 I 000001 ISTAPE    0010   000000 ISTPUP
0010   000001 IUNSYM    0000 I 012525 J         0000 I 012526 K         0000 I 012541 L         0000 I 012533 LOCND1
0000 I 012506 LOCOND    0000 R 000315 LDCOOD    0000 I 012507 LDIND     0000 I 007277 LONCDE    0000 L 000004 LNK
0000 L 000000 LOADS     0000 I 012542 M         0017 R 000567 MACDIF    0017 R 000000 MACH      0000 I 006063 MODES
0015 L 000001 MODPRT    0012 I 000000 MTAPE     0005 I 000010 NDELTA    0005 I 000000 NDF       0005 I 000007 NOTIME
0000 I 012543 NENG      0005   000011 NFC       0000 I 012532 NFILES    0005 I 000001 NFS       0005 I 000003 NLOND
0000 I 012544 NLINK     0000 I 012545 NLKF      0005 I 000002 NLS       0005 I 000005 NNM       0000 I 012240 NODEID
0005 I 000006 NODES     0000 I 000762 NODFIT    0000 I 012517 NOORB     0000 I 012520 NOSRML    0000 I 012521 NOSRMR
0000 I 012522 NOTANK    0005 I 000004 NTPTS     0000 I 012534 NUMBMD    0017   000566 NUMINC    0006 L 000003 ONEG
0007 I 000004 OPFCHK    0007 L 000003 OPLOD     0007 L 000001 OPMDA     0007 L 000000 OPRBA     0007 L 000002 OPTOTA
0015 L 000002 ORDER     0000   013330 OUTPT     0015 L 000003 PLOT      0011 L 000000 PLTMA     0011 L 000001 PLTMD
0017 R 000422 QPSI      0000 R 000430 SCALE     0017 R 000420 STARTM    0000 R 012513 STPSIZ    0010 L 000002 SYM
0000 R 012516 TILT      0017 R 000504 TIME      0011 R 000002 TITLE     0000 R 012523 TOL       0000 R 006001 TTABLE
0000 R 006227 VELOC     0000 R 012527 XCG       0000 R 006373 XMAT      0000 R 012510 XREF      0000 R 012530 YCG
0000 R 012511 YREF      0000 R 012531 ZCG       0006 L 000001 ZEROFC    0000 R 012512 ZREF
```

```
00101      1*          SUBROUTINE DREAD(REPORT)                                 DREAD001    000000
00101      2*    C                                                              DREAD002    000000
00101      3*    C     * * * * * * * * * * * * * * * * * * *                    DREAD003    000000
00101      4*    C     *  DREAD--CARD DATA READ SUBROUTINE  *                   DREAD004    000000
00101      5*    C     * * * * * * * * * * * * * * * * * * *                    DREAD005    000000
00101      6*    C                                                              DREAD006    000000
00103      7*          PARAMETER NIMAX=25,NFMAX=200,NLMAX=75,NTMAX=50                       000000
00104      8*          PARAMETER NNMAX=150,NDMAX=400,NMMAX=50                               000000
00105      9*          LOGICAL REPORT,SYM,ANTI,LOADS,FPLOT,ERROR                DREAD009    000000
00106     10*          LOGICAL CKOUT,FORTAP,MODPRT,ORDER,PLOT                   DREAD010    000000
00107     11*          LOGICAL OPRBA,OPMDA,OPTOTA,OPLOD                         DREAD011    000000
00110     12*          LOGICAL HOLDIT,ZEROFC,COMPIC,ONEG                        DREAD012    000000
```

ORIGINAL PAGE IS OF POOR QUALITY

28

```
00111          LOGICAL PLTMA,PLTMD,ALLMOD,CGCALC,ISAS                                       000000
00112          LOGICAL ENGID,EXTFOR,LNK                                                     000000
00113          LOGICAL FCFILE/.FALSE./, FOCOFL/.FALSE./                                     000000
00116          INTEGER FORMID, CNFGCK(20), OPFCHK                                           000000
00117          REAL LDCOOD(3,NIMAX)                                              DREAD015   000000
00120          REAL  MACH(NTMAX), MACDIF                                                    000000
00120    C                                                                       DREAD016   000000
00121          DIMENSION SCALE(NFMAX),FORXYZ(3,3),AFTXYZ(3,3),NODFIT(NIMAX,3)    DREAD017   000000
00122          DIMENSION F1(NTMAX),F2(NTMAX),F3(NTMAX),F4(NTMAX),F5(NTMAX),      DREAD018   000000
00122         1F6(NTMAX),F7(NTMAX),F8(NTMAX),F9(NTMAX),F10(NTMAX),F11(NTMAX),    DREAD019   000000
00122         2F12(NTMAX),F13(NTMAX),F14(NTMAX),F15(NTMAX),F16(NTMAX),F17(NTMAX),DREAD020   000000
00122         3F18(NTMAX),F19(NTMAX),F20(NTMAX),F21(NTMAX),F22(NTMAX),F23(NTMAX),DREAD021   000000
00122         4F24(NTMAX),F25(NTMAX),F26(NTMAX),F27(NTMAX),F28(NTMAX),F29(NTMAX),DREAD022   000000
00122         5F30(NTMAX),F31(NTMAX),F32(NTMAX),F33(NTMAX),F34(NTMAX),F35(NTMAX),DREAD023   000000
00122         6F36(NTMAX),F37(NTMAX),F38(NTMAX),F39(NTMAX),F40(NTMAX),F41(NTMAX),DREAD024   000000
00122         7F42(NTMAX),F43(NTMAX),F44(NTMAX),F45(NTMAX),F46(NTMAX),F47(NTMAX),DREAD025   000000
00122         8F48(NTMAX),F49(NTMAX),F50(NTMAX),TTABLE(NTMAX),FORMID(NFMAX)      DREAD026   000000
00122    C                                                                       DREAD027   000000
00123          DIMENSION MODES(NMMAX),DAMP(NMMAX),VELOC(NMMAX),DISPL(NMMAX)      DREAD028   000000
00123         1,XMAT(NNMAX,3),FTILE(2)                                                      000000
00123    C                                                                       DREAD030   000000
00124          DIMENSION LDNODE(NIMAX,NIMAX),IDACC(NLMAX),BGNTIM(11),            DREAD031   000000
00124         1 DELTIM(10),DMASS(NDMAX),IDXYZ(NDMAX),TITLE(12),NODEID(NNMAX)     DREAD032   000000
00124    C                                                                       DREAD033   000000
00125          DIMENSION ALPHA(NTMAX), BETA(NTMAX), DELTAE(NTMAX), DELTAR(NTMAX)            000000
00126          DIMENSION TIME(NTMAX), QPSI(NTMAX)                                           000000
00127          COMMON/ALOGIC/ERROR,CKOUT                                         DREAD034   000000
00130          COMMON/CASDAT/CTITLE(12)                                          DREAD035   000000
00131          COMMON/CDATA/NDF,NFS,NLS,NLOND,NTPTS,NNM,NODES,NDTIME,NDELTA,NFC             000000
00132          COMMON/CLOGIC/HOLDIT,ZEROFC,COMPIC,ONEG                           DREAD037   000000
00133          COMMON/OLOGIC/OPRBA,OPMDA,OPTOTA,OPLOD,OPFCHK                            D   000000
00134          COMMON/SYMLOG/ISYMOP,IUNSYM,SYM,ANTI,CGCALC                       DREAD039   000000
00135          COMMON/PLOGIC/PLTMA,PLTMD,TITLE                                   DREAD040   000000
00136          COMMON/SCRDRM/ MTAPE, IFORC, IPHIS, IDRUM, ISCRT, IFCSCR                     000000
00137          COMMON/IPTAPE/ICARDS,INPUT,IFIN,IFCTAP,IFCFIL                                000000
00140          COMMON/OPTAPE/IORDR,IRPT,IFPLOT,ILINKS                            DREAD043   000000
00141          COMMON/TLOGIC/FORTAP,MODPRT,ORDER,PLOT,FPLOT                      DREAD044   000000
00142          COMMON/OUTSAS/ISAS,ISTAPE                                                    000000
00143          COMMON /AERO / MACH, ALPHA, BETA, DELTAE, DELTAR, FCFILE, FOCOFL,            000000
00143         1  CNFGCK, STARTM, ENDTIM, QPSI, TIME, NUMINC, MACDIF, ALFDIF,                000000
00143         2  BETADF, DELEDF, DELRDF                                                     000000
00143    C                                                                       DREAD045   000000
00144          NAMELIST /IPGFP/ FORMID,TTABLE,F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,DREAD046   000000
00144         1F12,F13,F14,F15,F16,F17,F18,F19,F20,F21,F22,F23,F24,F25,F26,F27,  DREAD047   000000
00144         2F28,F29,F30,F31,F32,F33,F34,F35,F36,F37,F38,F39,F40,F41,F42,F43,  DREAD048   000000
00144         3F44,F45,F46,F47,F48,F49,F50,LDCOND,MODES,DAMP,LDNODE,LDCOOD,LDIND,DREAD049   000000
00144         4 IDACC,SCALE,XREF,YREF,ZREF,DISPL,VELOC,STPSIZ,BGNPRT,ENDPRT,     DREAD050   000000
00144         5BGNTIM,DELTIM,FORTAP,COMPIC,HOLDIT,ZEROFC,ALLMOD,ONEG,OPRBA,OPMDA,DREAD051   000000
00144         6 OPTOTA,OPLOD,MODPRT,CKOUT,ORDER,PLOT,PLTMD,PLTMA,FPLOT,          DREAD052   000000
00144         7 REPORT,CGCALC,SYM,ANTI,TILT,FORXYZ,AFTXYZ,NODFIT,NOORB,NOSRML,   DREAD053   000000
00144         8NOSRMR,NOTANK,ISAS,MACH,ALPHA,BETA,DELTAE,DELTAR,FCFILE,FOCOFL,              000000
00144         9  CNFGCK, TIME, QPSI, MACDIF, ALFDIF, BETADF,                                000000
00144         *  DELEDF, DELRDF, OPFCHK                                                     000000
00144    C                                                                       DREAD055   000000
00145          NAMELIST/OUTPT/LDCOND,XREF,YREF,ZREF,TILT,BGNPRT,ENDPRT,STPSIZ,   DREAD056   000000
00145         1BGNTIM,DELTIM,MODES,DAMP,DISPL,VELOC,FORMID,SCALE,IDACC,LDNODE,   DREAD057   000000
```

29

```
00145        3ANTI,CGCALC,CKOUT,COMPIC,FORTAP,FPLOT,HOLDIT,MODPRT,ONEG,OPLOD,   DREAD059   000000
00145        4OPMDA,OPRBA,OPTOTA,ORDER,PLOT,PLTMA,PLTMD,REPORT,SYM,ZEROFC       DREAD060   000000
00145        5,ISAS,MACH,ALPHA,BETA,DELTAE,DELTAR,FCFILE,FOCOFL,STARTM,ENDTIM,             000000
00145        6  OPFCHK                                                                     000000
00146         DATA FTILE/' FRISBE DATA'/                                                   000000
00150         HOLDIT=.FALSE.                                                    DREAD061   000000
00151         CKOUT=.FALSE.                                                     DREAD062   000000
00152         ZEROFC=.FALSE.                                                    DREAD063   000001
00153         COMPIC=.FALSE.                                                    DREAD064   000002
00154         ONEG=.FALSE.                                                      DREAD065   000003
00155         ISAS=.FALSE.                                                                 000004
00156         OPRBA=.TRUE.                                                      DREAD066   000005
00157         OPMDA=.TRUE.                                                      DREAD067   000007
00160         OPTOTA=.TRUE.                                                     DREAD068   000010
00161         OPLOD=.TRUE.                                                      DREAD069   000011
00162         OPFCHK = 0                                                                   000012
00163         PLTMA=.FALSE.                                                     DREAD070   000013
00164         PLTMD=.FALSE.                                                     DREAD071   000014
00165         FPLOT=.FALSE.                                                     DREAD072   000015
00166         REPORT=.FALSE.                                                    DREAD073   000016
00167         CGCALC=.FALSE.                                                    DREAD074   000017
00170         LOADS=.TRUE.                                                      DREAD075   000020
00171         ANTI=.FALSE.                                                      DREAD076   000021
00172          SYM=.FALSE.                                                      DREAD077   000022
00173         TOL=1.0E-11                                                       DREAD078   000023
00174         TILT=0.0                                                          DREAD079   000025
00175         LDINO=0                                                           DREAD080   000026
00176         XREF=0.0                                                          DREAD081   000027
00177         YREF=0.0                                                          DREAD082   000030
00200         ZREF=0.0                                                          DREAD083   000031
00201         MACDIF = .01                                                                 000032
00202         ALFDIF = .05                                                                 000034
00203         BETADF = .05                                                                 000036
00204         DELEDF = .10                                                                 000037
00205         DELRDF = .15                                                                 000041
00206         DO 10 I=1,NDMAX                                                   DREAD084   000046
00211         DMASS(I)=0.0                                                      DREAD085   000046
00212         IOXYZ(I)=0                                                        DREAD086   000046
00213      10 CONTINUE                                                          DREAD087   000053
00215         DO 20 I=1,NLMAX                                                   DREAD088   000053
00220         IDACC(I)=0                                                        DREAD089   000053
00221      20 CONTINUE                                                          DREAD090   000057
00223         DO 30 I=1,NFMAX                                                   DREAD091   000057
00226         SCALE(I)=1.0                                                      DREAD092   000057
00227         FORMID(I)=0                                                       DREAD093   000060
00230      30 CONTINUE                                                          DREAD094   000070
00232         DO 50 I=1,NIMAX                                                   DREAD095   000070
00235         DO 40 J=1,3                                                       DREAD096   000070
00240         NODFIT(I,J)=0                                                     DREAD097   000070
00241         LDCOOD(J,I)=0.0                                                   DREAD098   000070
00242      40 CONTINUE                                                          DREAD099   000074
00244         DO 50 K=1,NIMAX                                                   DREAD100   000074
00247         LONODE(K,I)=0                                                     DREAD101   000074
00250      50 CONTINUE                                                          DREAD102   000104
00253         DO 60 I=1,NTMAX                                                   DREAD103   000104
00256         MACH(I) = 0.0                                                                000104
00257         ALPHA(I) = 0.0                                                               000104
```

ORIGINAL PAGE IS OF POOR QUALITY

-30-

```
00260        BETA(I) = 0.0                                      000105
00261        DELTAE(I) = 0.0                                    000106
00262        DELTAR(I) = 0.0                                    000107
00263        TTABLE(I)=0.0                           DREAD104   000110
00264     60 CONTINUE                                DREAD105   000115
00266        DO 70 I=1,NMMAX                         DREAD106   000115
00271        DISPL(I)=0.0                            DREAD107   000115
00272        VELOC(I)=0.0                            DREAD108   000115
00273        MODES(I)=0                              DREAD109   000116
00274        DAMP(I)=0.0                             DREAD110   000117
00275     70 CONTINUE                                DREAD111   000124
00277        DO 80 I=1,10                            DREAD112   000124
00302        BGNTIM(I)=0.0                           DREAD113   000124
00303        DELTIM(I)=0.0                           DREAD114   000124
00304     80 CONTINUE                                DREAD115   000132
00306        DO 90 I=1,3                             DREAD116   000132
00311        DO 90 J=1,3                             DREAD117   000132
00314        FORXYZ(I,J)=0.0                         DREAD118   000132
00315        AFTXYZ(I,J)=0.0                         DREAD119   000132
00316     90 CONTINUE                                DREAD120   000140
00321        J=1                                     DREAD121   000140
00321  C                                                        000140
00321  C       THE NAMELIST DATA IS READ IN AT THIS POINT.      000140
00321  C                                                        000140
00322        READ (ICARDS,IPGFP)                     DREAD122   000142
00325    100 CONTINUE                                DREAD123   000147
00326        J=J+1                                   DREAD124   000147
00327        IF(J.GT.11) GO TO 110                   DREAD125   000151
00331        IF(BGNTIM(J).GT.0.0) GO TO 100          DREAD126   000155
00333    110 CONTINUE                                DREAD127   000163
00334        STARTM = BGNTIM(1)                                 000163
00335        ENDTIM = BGNTIM(J-1)                               000164
00336        NDELTA=J-1                              DREAD128   000167
00337        NDTIME=J-2                              DREAD129   000172
00340        J=1                                     DREAD130   000175
00341        IF(.NOT.FORTAP)  GO TO 120              DREAD131   000177
00343        J=TTABLE(1)+1                           DREAD132   000201
00344        GO TO 130                               DREAD133   000212
00345    120 CONTINUE                                DREAD134   000214
00346        J=J+1                                   DREAD135   000214
00347        IF(J.GT.NTMAX) GO TO 130                DREAD136   000216
00351        IF(ABS(TTABLE(J)).GT.TOL) GO TO 120     DREAD137   000222
00353    130 CONTINUE                                DREAD138   000231
00354        NTPTS=J-1                               DREAD139   000231
00355        XCG=XREF                                DREAD140   000233
00356        YCG=YREF                                DREAD141   000235
00357        ZCG=ZREF                                DREAD142   000237
00360        I=1                                     DREAD143   000241
00361    140 CONTINUE                                DREAD144   000244
00362        I=I+1                                   DREAD145   000244
00363        IF(I.GT.NLMAX) GO TO 150                DREAD146   000246
00365        IF(IDACC(I).NE.0) GO TO 140             DREAD147   000252
00367    150 CONTINUE                                DREAD148   000256
00370        NLS=I-1                                 DREAD149   000256
00371        REWIND INPUT                            DREAD150   000260
00372        READ (INPUT) NFILES                     DREAD151   000263
00375        IF(LDCOND.LT.2.OR.LDCOND.GT.NFILES) GO TO 160   DREAD152   000270
```

```
00377         GO TO 190                                                              DREAD153   000306
00400     160 CONTINUE                                                               DREAD154   000310
00401         PRINT 170                                                              DREAD155   000310
00403     170 FORMAT(1H1,37X,'****ERROR ENCOUNTERED IN INPUT DATA*****',/)           DREAD156   000313
00404         PRINT 180, LDCOND,NFILES                                               DREAD157   000313
00410     180 FORMAT(10X,'LDCOND=',I3,2X,'NFILES=',I3)                               DREAD158   000321
00411         ERROR=.TRUE.                                                           DREAD159   000321
00412         RETURN                                                                 DREAD160   000323
00413     190 CONTINUE                                                               DREAD161   000327
00414         LDCND1=LDCOND-1                                                        DREAD162   000327
00415         CALL FILE(INPUT,LDCND1)                                                DREAD163   000331
00416         READ (INPUT) (TITLE(I),I=1,12)                                         DREAD164   000335
00424         PRINT 200, TITLE                                                       DREAD165   000346
00427     200 FORMAT(1H1,15X,'MODAL TAPE TITLE IS ****',12A6,'*****')                DREAD166   000355
00430         PRINT 210, CTITLE                                                      DREAD167   000355
00433     210 FORMAT(///,18X,'CASE TITLE IS *****',12A6,'*****')                     DREAD168   000364
00434         READ (INPUT) NDF,NUMBMD,(IDXYZ(I),I=1,NDF),NODES,(NODEID(I),I=1,       DREAD169   000364
00434        1NODES),((XMAT(I,J),J=1,3),I=1,NODES),(DMASS(I),I=1,NDF),GMASS          DREAD170   000364
00465         TILT=TILT/57.29578                                                     DREAD171   000436
00466         I=0                                                                    DREAD172   000441
00467         IF(.NOT.ALLMOD) GO TO 230                                              DREAD173   000442
00471         NNM=MODES(1)                                                           DREAD174   000444
00472         DMP=DAMP(1)                                                            DREAD175   000446
00473         IC=0                                                                   DREAD176   000450
00474         IF(DAMP(2).GT.0.0) IC=1                                                DREAD177   000451
00476         IF(NNM.GT.NMMAX) NNM=NMMAX                                             DREAD178   000457
00500         IF(NNM.GT.NUMBMD) NNM=NUMBMD                                           DREAD179   000472
00502         DO 220 I=1,NNM                                                         DREAD180   000501
00505         MODES(I)=I                                                             DREAD181   000506
00506         IF(IC.NE.0) GO TO 220                                                  DREAD182   000510
00510         DAMP(I)=DMP                                                            DREAD183   000512
00511     220 CONTINUE                                                               DREAD184   000516
00513         GO TO 260                                                              DREAD185   000516
00514     230 CONTINUE                                                               DREAD186   000520
00515         I=I+1                                                                  DREAD187   000520
00516         IF(I.GT.NMMAX) GO TO 240                                               DREAD188   000522
00520         IF(MODES(I).NE.0) GO TO 230                                            DREAD189   000526
00522     240 CONTINUE                                                               DREAD190   000532
00523         NNM=I-1                                                                DREAD191   000532
00524         IF(NNM.LE.NUMBMD) GO TO 260                                            DREAD192   000534
00526         DO 250 ICZ=1,NNM                                                       DREAD193   000540
00531         IF(MODES(ICZ).GT.NUMBMD) MODES(ICZ)=0                                  DREAD194   000545
00533     250 CONTINUE                                                               DREAD195   000554
00535         NNM=NUMBMD                                                             DREAD196   000554
00536     260 CONTINUE                                                               DREAD197   000557
00537         WRITE (6,OUTPT)                                                        DREAD198   000557
00542         I=1                                                                    DREAD199   000562
00543     270 CONTINUE                                                               DREAD200   000565
00544         I=I+1                                                                  DREAD201   000565
00545         IF(I.GT.NFMAX) GO TO 280                                               DREAD202   000567
00547         IF(FORMID(I).NE.0) GO TO 270                                           DREAD203   000573
00551     280 CONTINUE                                                               DREAD204   000577
00552         NFS=I-1                                                                DREAD205   000577
00553         I=1                                                                    DREAD206   000601
00554     290 CONTINUE                                                               DREAD207   000604
00555         I=I+1                                                                  DREAD208   000604
00556         IF(I.GT.NIMAX) GO TO 300                                               DREAD209   000606
```

ORIGINAL PAGE IS OF POOR QUALITY

```
00560        IF(LDNODE(1,I).NE.0) GO TO 290                                  DREAD210   000612
00562    300 CONTINUE                                                        DREAD211   000617
00563        NLDND=I-1                                                       DREAD212   000617
00564        IF(NLDND.EQ.1.AND.LDNODE(1,1).EQ.0) LOADS=.FALSE.                DREAD213   000621
00566        PRINT 310                                                       DREAD214   000650
00570    310 FORMAT(1H1,50X,'CO-ORDINATE DATA',//,26X,'NODE',3X,'NO.',17X,'X',  DREAD215   000660
00570       115X,'Y',15X,'Z',/)                                              DREAD216   000660
00571        DO 330 K=1,NODES                                                DREAD217   000660
00574        PRINT 320, NODEID(K),K,(XMAT(K,L),L=1,3)                        DREAD218   000660
00604    320 FORMAT(25X,I5,2X,'(',I3,')',5X,3F15.2)                          DREAD219   000677
00605    330 CONTINUE                                                        DREAD220   000677
00607        PRINT 340                                                       DREAD221   000677
00611    340 FORMAT(1H1,45X,'MASS MODEL',//,18X,'MASSES ARE GIVEN IN LB-SEC 2/IDREAD222   000703
00611       IN AND INERTIAS ARE GIVEN IN LB-IN-SEC 2',//)                    DREAD223   000703
00612        PRINT 350, (IDXYZ(J),J,DMASS(J),J=1,NDF)                        DREAD224   000703
00622    350 FORMAT(4(3X,I5,'(',I3,')',3X,1PE10.3))                          DREAD225   000720
00623        REWIND MTAPE                                                    DREAD226   000720
00624        REWIND IPHIS                                                    DREAD227   000723
00625        WRITE (IPHIS) NUMBMD,GMASS,(DAMP(K),K=1,NNM),(MODES(I),I=1,NNM),  DREAD228   000726
00625       1 (TTABLE(L),L=1,NTMAX)                                          DREAD229   000726
00645        WRITE (IPHIS) (SCALE(I),I=1,NFS)                                DREAD230   000753
00653        IF(FORTAP) GO TO 370                                            DREAD231   000764
00655        DO 360 I=1,NTMAX                                                DREAD232   000771
00660        WRITE (IPHIS) (F1(I),F2(I),F3(I),F4(I),F5(I),F6(I),F7(I),F8(I),  DREAD233   000771
00660       1 F9(I),F10(I),F11(I),F12(I),F13(I),F14(I),F15(I),F16(I),F17(I),  DREAD234   000771
00660       2F18(I),F19(I),F20(I),F21(I),F22(I),F23(I),F24(I),F25(I),F26(I),  DREAD235   000771
00660       3F27(I),F28(I),F29(I),F30(I),F31(I),F32(I),F33(I),F34(I),F35(I),  DREAD236   000771
00660       4F36(I),F37(I),F38(I),F39(I),F40(I),F41(I),F42(I),F43(I),F44(I),  DREAD237   000771
00660       5F45(I),F46(I),F47(I),F48(I),F49(I),F50(I))                      DREAD238   000771
00744    360 CONTINUE                                                        DREAD239   001060
00746    370 CONTINUE                                                        DREAD240   001060
00747        END FILE IPHIS                                                  DREAD241   001060
00750        REWIND IPHIS                                                    DREAD242   001062
00751        WRITE (MTAPE) (IDXYZ(I),I=1,NDF),(IDACC(K),K=1,NLS),(FORMID(L),L= DREAD243   001065
00751       1 1,NFS)                                                         DREAD244   001065
00767        WRITE (MTAPE) XREF,YREF,ZREF,(NODEID(I),I=1,NODES),((XMAT(I,J),  DREAD245   001141
00767       1 J=1,3),I=1,NODES),(DMASS(K),K=1,NDF)                           DREAD246   001141
01013        WRITE (MTAPE) ((LDCOOD(I,J),I=1,3),J=1,NIMAX),((LDNODE(K,M),K=1,  DREAD247   001174
01013       1 NIMAX),M=1,NLDND),LDIND                                        DREAD248   001174
01034        WRITE (MTAPE) ((NODFIT(I,J),I=1,NIMAX),J=1,3),((FORXYZ(I,J),I=1,3)DREAD249   001214
01034       1,J=1,3),((AFTXYZ(I,J),I=1,3),J=1,3)                             DREAD250   001214
01063        WRITE (MTAPE) (DAMP(I),I=1,NNM),(DISPL(I),I=1,NNM),(VELOC(K),K=1,  DREAD251   001237
01063       1 NNM),GMASS                                                     DREAD252   001237
01102        WRITE (MTAPE) STPSIZ,BGNPRT,ENDPRT,(BGNTIM(I),I=1,NDELTA),       DREAD253   001264
01102       1(DELTIM(K),K=1,NDTIME)                                          DREAD254   001264
01117        WRITE (MTAPE) REPORT,LOADS                                      DREAD255   001305
01123        WRITE (MTAPE) TILT                                              DREAD256   001313
01126        END FILE MTAPE                                                  DREAD257   001320
01127        REWIND MTAPE                                                    DREAD258   001323
01130        REWIND ILINKS                                                   DREAD259   001326
01131        WRITE (ILINKS) NOORB,NOSRML,NOSRMR,NOTANK                       DREAD260   001331
01137        WRITE (ILINKS) ((FORXYZ(I,J),I=1,3),J=1,3),((AFTXYZ(I,J),I=1,3),  DREAD261   001341
01137       1 J=1,3)                                                         DREAD262   001341
01157        END FILE ILINKS                                                 DREAD263   001357
01160        REWIND ILINKS                                                   DREAD264   001362
                                                                                        001365
```

```
01164        WRITE (ISTAPE) (FTILE(I),I=1,2),(TITLE(J),J=1,10),(CTITLE(K),K=1,      001375
01164       112)                                                                     001375
01202        ENGID=.TRUE.                                                            001420
01203        EXTFOR=.TRUE.                                                           001422
01204        LNK=.FALSE.                                                             001423
01205        NENG=0                                                                  001424
01206        NLINK=0                                                                 001425
01207    380 CONTINUE                                                                001427
01210        NLINK=NLINK+1                                                           001427
01211        IF(NODFIT(1,NLINK).GT.0) GO TO 380                                      001431
01213        NLINK=NLINK-1                                                           001436
01214        NLKF=NLINK+12                                                           001441
01215        WRITE(ISTAPE) ENGID,EXTFOR,NENG,NFS,NLKF,NNM,LNK,BGNTIM(1),             001443
01215       1BGNTIM(NDELTA)                                                          001443
01230        WRITE (ISTAPE) (FORMID(I),I=1,NFS)                                      001464
01236        IF(NNM.LE.0) GO TO 375                                                  001475
01240        WRITE (ISTAPE) (MODES(K),K=1,NNM)                                       001501
01246    375 CONTINUE                                                                001515
01247        IF(NLINK.LE.0) GO TO 385                                                001515
01251        WRITE (ISTAPE) ((FORXYZ(I,J),I=1,3),(AFTXYZ(K,J),K=1,3),J=1,NLINK)      001520
01266    385 CONTINUE                                                                001543
01267        RETURN                                                  DREAD265        001543
01270        END                                                     DREAD266        001604
```

END OF COMPILATION: NO DIAGNOSTICS.

@HDG,P •••••• FCREAD ••••••

34

4.4.2 Subroutine FCREAD

The purpose of subroutine FCREAD is to read the flight conditions file.

STORAGE:

IBANK	1017_8
DBANK	11746_8

PARAMETERS:

	Size	Description
NTMAX	50	Number of time points - max.
NNMAX	150	Number of nodes - max.
NMMAX	50	Number of modes - max.
MAXENG	3	Number of engines - max.
MAXEID	6*MAXENG	Number of engine ID's - max.
MAXFEX	6*NNMAX	Number of external forces - max.
MAXLNK	6	Number of links - max.
MAXEXY	3*MAXENG	Number of engine location dimension - max.
MAXFXY	3*NNMAX	Number of force location dimensions - max.
MAXLID	6*MAXLNK	Number of link forces - max.
MAXLXY	3*MAXLNK	Number of link location dimensions - max.

INPUT VARIABLES:

FCFILE	L	If true read flight conditions tape
FOCOFL	L	If true read force coefficient file
STARTM	R	Start time for the run
ENDTIM	R	End time for the run
MACDIF	R	Accuracy requirement for MACH
ALFDIF	R	Accuracy requirement for ALPHA
BETADF	R	Accuracy required for BETA
DELEDF	R	Accuracy required for DELTAE
DELRDF	R	Accuracy required for DELTAR
IFCTAP	I	Unit number of flight conditions tape
IFCFIL	I	Unit number of force coefficient file
OPFCHK	I	Printout control

OUTPUT VARIABLES:

	Type	Size	Description
MACH	R	NTMAX	MACH number array
ALPHA	R	NTMAX	Angle of attack (ALPHA) array
BETA	R	NTMAX	Yaw angle (BETA) array
DELTAE	R	NTMAX	Elevon deflection array
DELTAR	R	NTMAX	Rudder deflection array
CNFGCK	A	20	Configuration description
QPSI	R	NTMAX	Dynamic pressure array
TIME	R	NTMAX	Flight condition time array
NDMINC	I	1	Number of flight conditions stored

TAPE AND OTHER MASS STORAGE INPUT:

Input to this subroutine includes flight condition file data containing previously described variables. The data is read from unit IFCTAP which is numerically designated as unit 19. This unit must contain the tape containing the desired Batch Flight Conditions File . The format of this file is described in Appendix A.

ERROR MESSAGES AND WARNINGS:

- "THE REQUESTED START TIME IS NOT ON THIS FILE. SOME RECORDS MAY NOT BE PRESENT".

The run time is automatically set to the start time of the flight conditions file.

-36-

• "THE REQUESTED STOP TIME IS NOT ON THIS TAPE. SOME RECORDS MAY NOT BE PRESENT".

The stop time is automatically set to the end time on the flight conditions tape.

PRINTED OUTPUT:

The printed output from FCREAD is controlled by input variable OPFCHK:

1. If OPFCHK = 0 Only the title of the flight conditions tape is output
2. If OPFCHK = 1 No additional output
3. If OPFCHK = 2 Namelist INPUT1 and INREPT are printed. INREPT each time a flight condition is read.
4. If OPFCHK > 2 Namelist LKMODS is printed for each flight condition read.

Refer to Section 4.3.1 for namelist description.

FLOW CHART:

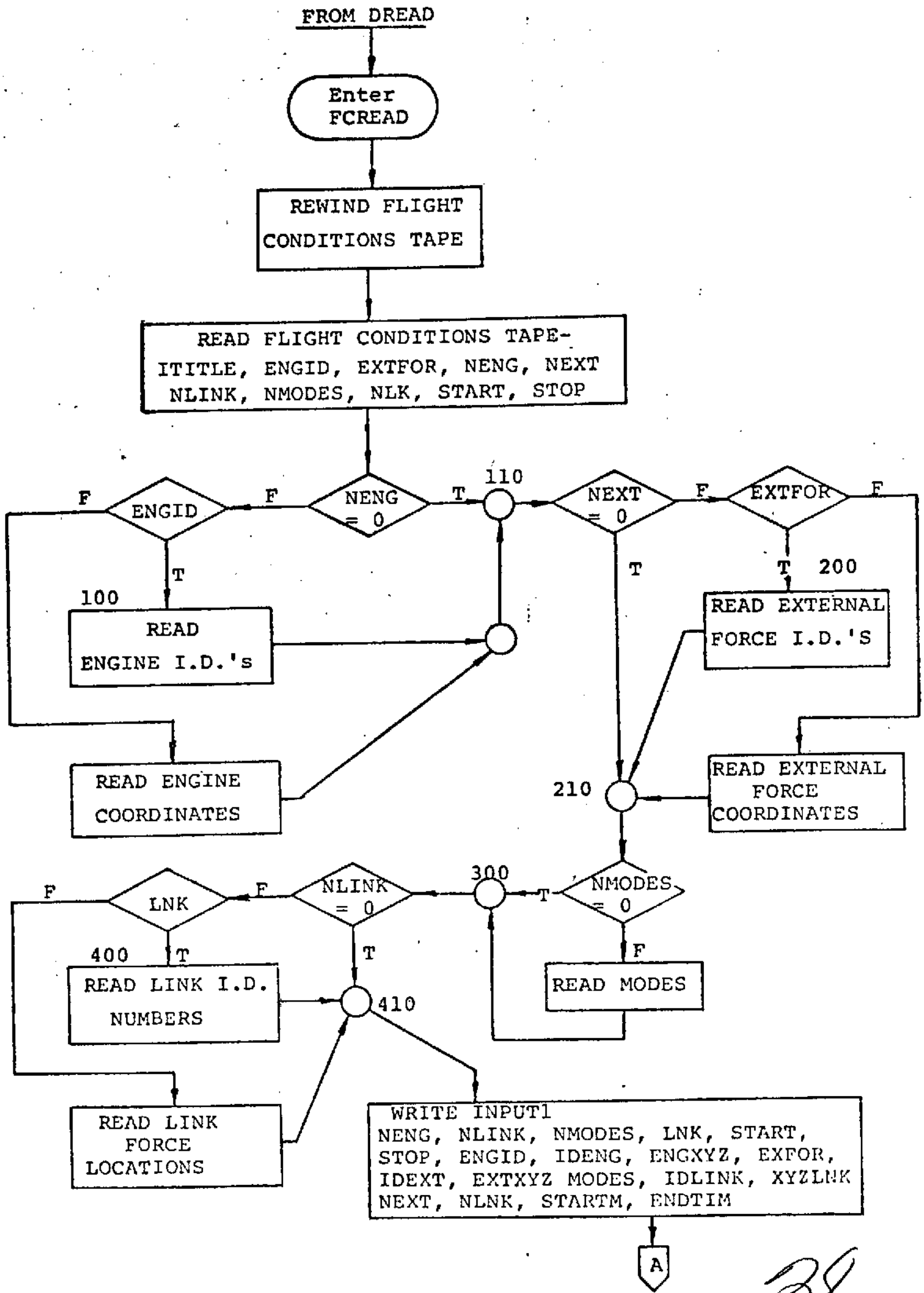
A detailed flow chart of subroutine FCREAD is shown in 4.4.2.1.

LISTING:

A listing of subroutine FCREAD is shown in 4.4.2.2.

-37-

4.4.2.1 FLOW CHART OF SUBROUTINE FCREAD



38

FIGURE 4-4
SUBROUTINE FCREAD FUNCTIONAL FLOW

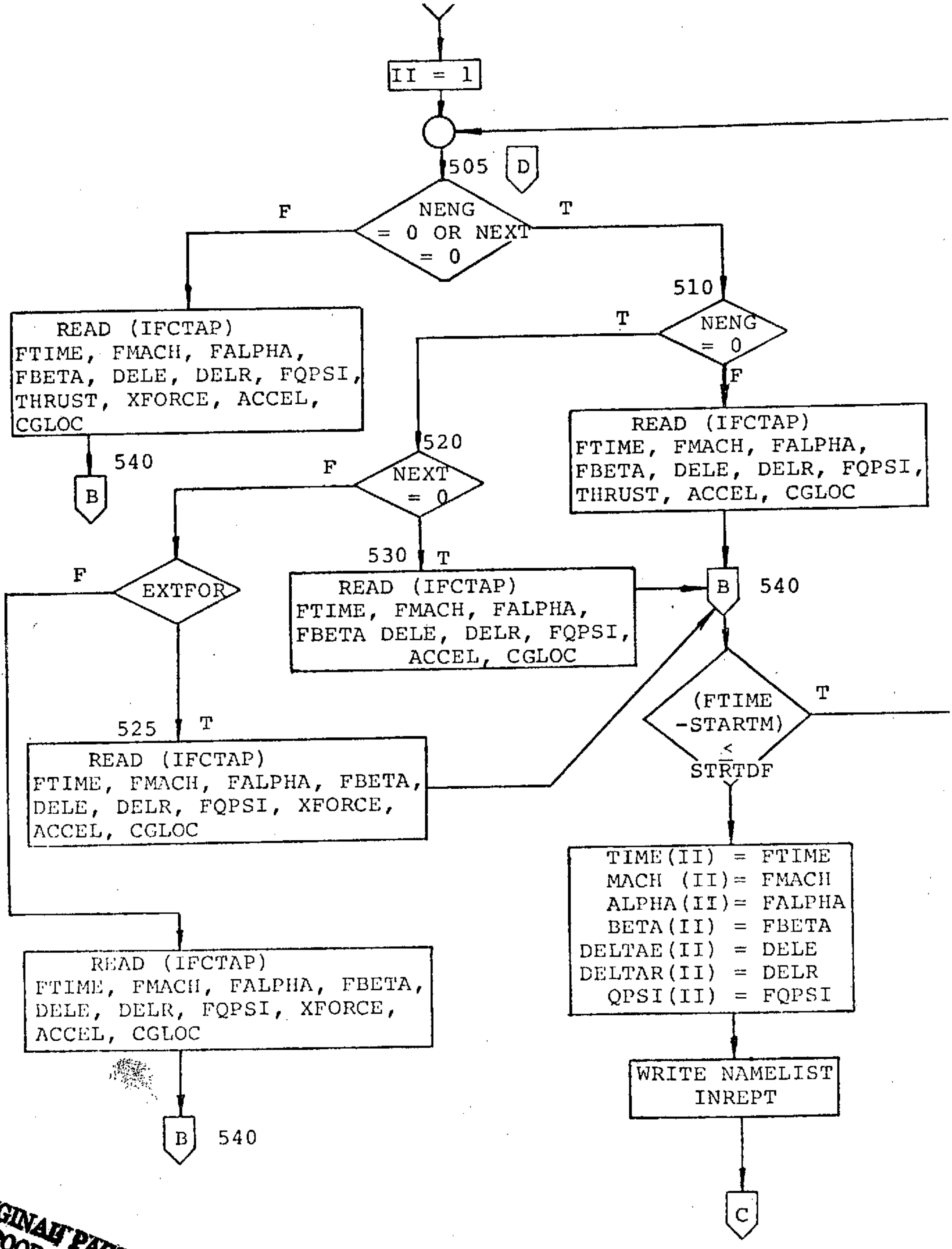


FIGURE 4-4 (cont'd)
SUBROUTINE FCREAD FUNCTIONAL FLOW

39

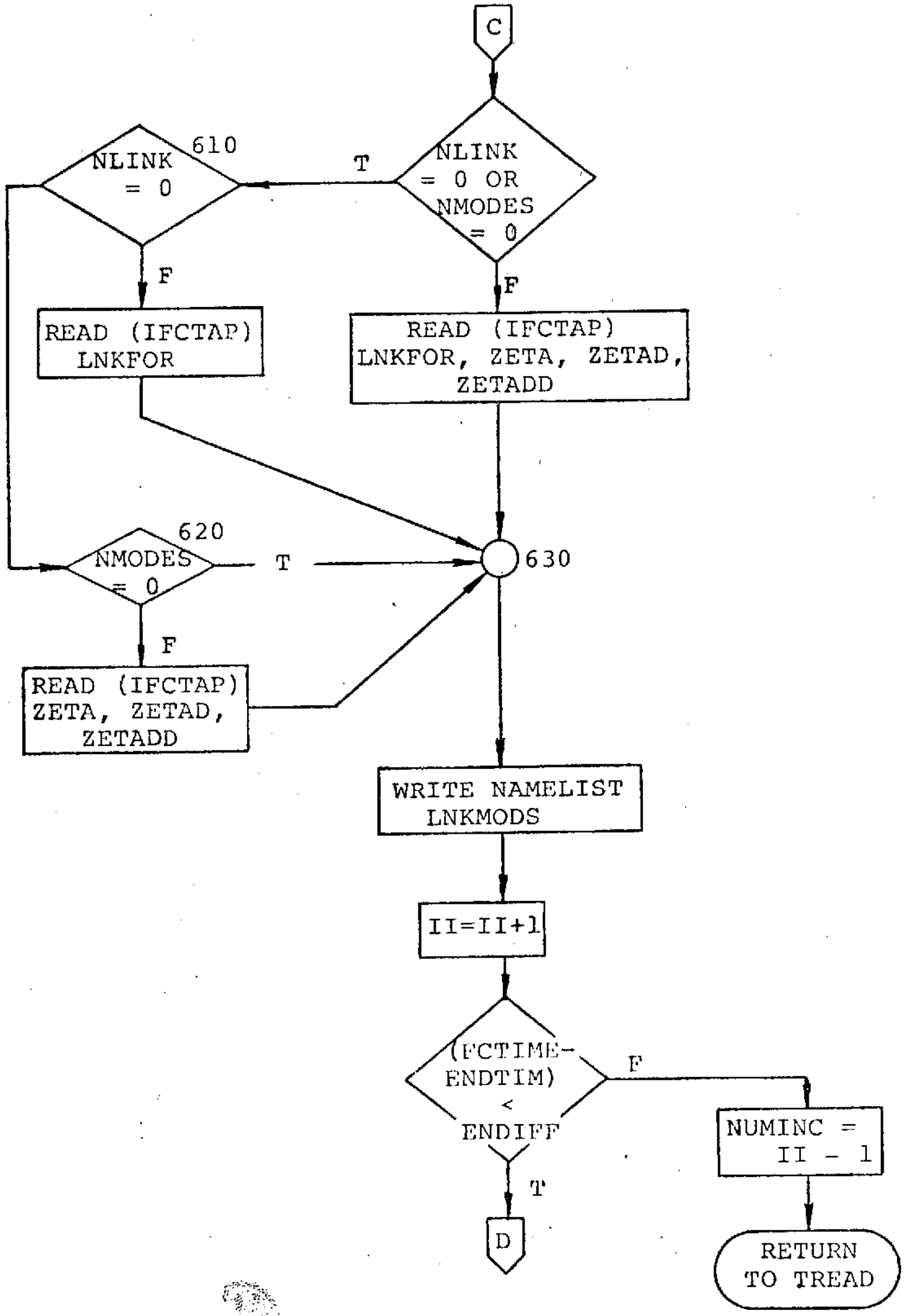


FIGURE 4-4 (cont'd)
SUBROUTINE FCREAD FUNCTIONAL FLOW

40

```
@FOR,S P,FCREAD,FCREAD
FOR SE1X-11/19/74-17:07:47 (24,)

   SUBROUTINE FCREAD     ENTRY POINT 001006

   STORAGE USED: CODE(1) 001017; DATA(0) 011746; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   AERO   000574
    0004   IPTAPE 000005
    0005   OLOGIC 000005

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006   NREA$
    0007   NRBU$
    0010   NIO1$
    0011   NIO2$
    0012   NWDU$
    0013   NWNL$
    0014   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0001     000142 100L      0001     000155 110L      0001     000074 120L      0001     000013 126G      0001     000103 130L
   0001     000111 140L      0001     000046 147G      0001     000120 150L      0001     000177 200L      0001     000135 206G
   0001     000135 210G      0001     000212 210L      0001     000151 220G      0001     000172 233G      0001     000172 235G
   0001     000206 245G      0001     000225 256G      0001     000246 271G      0001     000246 273G      0001     000231 300L
   0001     000262 303G      0001     000334 331G      0001     000334 333G      0001     000341 340G      0001     000341 342G
   0001     000346 347G      0001     000353 353G      0001     000401 373G      0001     000401 375G      0001     000253 400L
   0001     000406 402G      0001     000413 406G      0001     000266 410L      0001     000452 431G      0001     000452 433G
   0001     000457 440G      0001     000464 444G      0001     000507 462G      0001     000521 467G      0001     000526 473G
   0000     011655 5000F     0000     011675 5001F     0000     011651 5010F     0001     000301 505L      0001     000360 510L
   0001     000547 511G      0001     000554 515G      0001     000420 520L      0001     000471 525L      0001     000533 530L
   0001     000560 540L      0001     000642 543G      0001     000642 545G      0001     000647 552G      0001     000654 556G
   0001     000661 562G      0001     000700 573G      0001     000700 575G      0001     000717 607G      0001     000666 610L
   0001     000724 613G      0001     000731 617G      0001     000705 620L      0001     000735 630L      0000 R 011150 ACCEL
   0003     000570 ALFDIF    0003 R 000062 ALPHA     0003 R 000144 BETA      0003     000571 BETADF    0000 R 011156 CGLOC
   0003 I 000374 CNFGCK    0000 R 011444 DELE      0003     000572 DELEDF    0000 R 011445 DELR      0003     000573 DELRDF
   0003 R 000226 DELTAE    0003 R 000310 DELTAR    0000 R 011450 ENDIFF    0003 R 000421 ENDTIM    0000 R 011456 ENDTM
   0000 L 000000 ENGID     0000 R 000115 ENGXYZ    0000 L 000001 EXTFOR    0000 R 001754 EXTXYZ    0000 R 011442 FALPHA
   0000 R 011443 FBETA     0003     000372 FCFILE    0000 R 011441 FMACH     0003     000373 FOCUFL    0000 R 011446 FQPSI
   0000 R 011440 FTIME     0000 I 011451 I         0004     000000 ICARDS    0000 I 000126 IDENG     0000 I 000150 IDEXT
   0000 I 007170 IDLINK    0004     000004 IFCFIL    0004 I 000003 IFCTAP    0004     000002 IFIN      0000 I 011460 II
   0000     011724 INJP$     0004     000001 INPUT     0000     011467 INPUT1    0000     011566 INREPT    0000 I 000003 ITITLE
   0000 I 011457 K         0000     011633 LKMODS    0000 L 000002 LNK       0000 I 011407 LNKFOR    0003 R 000567 MACDIF
   0003 R 000000 MACH      0000 I 000033 MODES     0000 I 011431 NENG      0000 I 011436 NEXT      0000 I 011432 NLINK
   0000 I 011437 NLNK      0000 I 011433 NMODES    0000 I 011454 NUMENG    0003 I 000566 NUMINC    0000 I 011453 NUMLNK
   0000 I 011452 NUMAF     0005 I 000004 OPFCHK    0005     000003 OPLOD     0005     000001 OPMDA     0005     000000 OPRDA
   0005     000002 OPTOTA    0003 R 000422 QPSI      0000 R 011434 START     0003 R 000420 STARTM    0000 R 011435 STOP
   0000 R 011447 STRTDF    0000 R 011455 STRTM     0000 R 007322 THRUST    0003 R 000504 TIME      0000 R 007344 XFORCE
   0000 R 007234 XYZLNK    0000 R 011161 ZETA      0000 R 011243 ZETAD     0000 R 011325 ZETADD
```

```
00101              SUBROUTINE FCREAD                                                      000000
00103              PARAMETER  NTMAX=50,  NNMAX=150                                        000000
00104              PARAMETER    NNMAX=50,MAXENG=3,     MAXEID=6*MAXENG,                   000000
00104             1  MAXFEX=6*NNMAX,MAXLNK=6,MAXEXY=3*MAXENG,MAXFID=6*NNMAX,              000000
00104             2  MAXFXY=3*NNMAX,MAXLID=6*MAXLNK,MAXLXY=3*MAXLNK,                      000000
00105              REAL  MACH(NTMAX), MACDIF                                              000000
00106              INTEGER CNFGCK(20), OPFCHK                                             000000
00107              LOGICAL  ENGID, EXTFOR, LNK                                            000000
00110              DIMENSION ITITLE(24),MODES(NMMAX),ENGXYZ(3,MAXENG),IDENG(MAXEID),      000000
00110             1  IDEXT(MAXFID),EXTXYZ(3,MAXFEX),IDLINK(MAXLID),XYZLNK(3,MAXLXY),      000000
00110             2  THRUST(6,MAXENG), XFORCE(6,NNMAX), ACCEL(6), CGLOC(3),               000000
00110             3  ZETA(NMMAX), ZETAD(NMMAX), ZETADD(NMMAX),                            000000
00110             4   LNKFOR(3,MAXLNK)                                                    000000
00111              COMMON /AERO / MACH, ALPHA, BETA, DELTAE, DELTAR, FCFILE, FOCOFL,      000000
00111             1  CNFGCK, STARTM, ENDTIM, QPSI, TIME, NUMINC, MACDIF, ALFDIF,          000000
00111             2  BETADF, DELEDF, DELRDF                                               000000
00112              COMMON/IPTAPE/ICARDS,INPUT,IFIN,IFCTAP,IFCFIL                          000000
00113              COMMON/OLOGIC/OPRBA,OPMDA,OPTOTA,OPLOD,OPFCHK                          000000
00114              DIMENSION TIME(NTMAX), QPSI(NTMAX)                                     000000
00115              DIMENSION ALPHA(NTMAX), BETA(NTMAX), DELTAE(NTMAX), DELTAR(NTMAX)      000000
00116              NAMELIST/INPUT1/  NENG,NLINK,NMODES,LNK,START,STOP,ENGID,IDENG,        000000
00116             1  ENGXYZ,EXTFOR,IDEXT,EXTXYZ,MODES,IDLINK,XYZLNK,NEXT,NLNK             000000
00116             2  ,STARTM, ENDTIM                                                      000000
00117              NAMELIST/INREPT/ FTIME,FMACH,FALPHA,FBETA,DELE,DELR,FQPSI,THRUST,      000000
00117             1  XFORCE,ACCEL,CGLOC                                                   000000
00120              NAMELIST/LKMODS/  ZETADD,ZETAD,ZETA,LNKFOR,                            000000
00121              REWIND IFCTAP                                                          000000
00122              STRTDF = -.001                                                         000002
00123              ENDIFF = .001                                                          000004
00124              READ(IFCTAP) (ITITLE(I),I = 1,24)                                      000005
00132              READ (IFCTAP) ENGID,EXTFOR,NENG,NEXT,NLINK,NMODES,LNK,START,STOP       000016
00145              WRITE (6,5010) (ITITLE(I), I=1,24)                                     000036
00153      5010    FORMAT (1H1, 10X,12A6/8X,12A6)                                         000051
00154              NUMXF = NEXT / 6                                                       000051
00155              NUMLNK = NLINK / 6                                                     000054
00156              NUMENG = NENG / 6                                                      000056
00157              NLNK = NLINK / 6                                                       000061
00157      C       STRTM IS THE START TIME OF THE CONDITIONS ON THE FILE BEING INPUT      000061
00157      C       ENDTM IS THE END TIME OF THE CONDITION S BEING INPUT                   000061
00160              STRTM = START                                                          000062
00161              ENDTM = STOP                                                           000064
00162              IF (STRTM .GT. STARTM)  GO TO 120                                      000066
00164              GO TO 130                                                              000072
00165       120    WRITE (6,5000)                                                         000074
00167              STRTM = STARTM                                                         000100
00170      5000    FORMAT (//5X,'THE REQUESTED START TIME IS NOT ON THIS FILE.'/          000103
00170             1  4X,'SOME RECORDS MAY NOT BE PRESENT.')                               000103
00171       130    IF (ENDTM .LT. ENDTIM)   GO TO 140                                     000103
00173              GO TO 150                                                              000107
00174       140    WRITE (6,5001)                                                         000111
00176              ENDTM = ENDTIM                                                         000115
00177      5001    FORMAT (//5X,'THE REQUESTED STOP TIME IS NOT ON THIS FILE.'/           000120
00177             1  4X,'SOME RECORDS MAY NOT BE PRESENT.')                               000120
00200       150    IF (NENG .EQ. 0)   GO TO 110                                           000120
00202              IF (ENGID)   GO TO 100                                                 000121
```

ORIGINAL PAGE IS OF POOR QUALITY

```
00215                  GO TO 110                                                          000140
00216         100      READ (IFCTAP)  (IDENG(I),I=1,NENG)                                 000142
00224         110      CONTINUE                                                           000155
00225                  IF (NEXT .EQ. 0)   GO TO 210                                       000155
00227                  IF (EXTFOR)   GO TO 200                                            000156
00231                  READ (IFCTAP)  ((EXTXYZ(K,I),K=1,3),I=1,NUMXF)                     000163
00242                  GO TO 210                                                          000175
00243         200      READ (IFCTAP)  (IDEXT(I),I=1,NEXT)                                 000177
00251         210      CONTINUE                                                           000212
00252                  IF (NMODES .LE. 0)   GO TO 300                                     000212
00254                  READ (IFCTAP)  (MODES(I),I=1,NMODES)                               000215
00262         300      CONTINUE                                                           000231
00263                  IF (NLINK .EQ. 0)   GO TO 410                                      000231
00265                  IF (LNK)   GO TO 400                                               000232
00267                  READ (IFCTAP) ((XYZLNK(K,I),K=1,3),I=1,NUMLNK)                     000237
00300                  GO TO 410                                                          000251
00301         400      READ (IFCTAP)  (IDLINK(I),I=1,NLINK)                               000253
00307         410      CONTINUE                                                           000266
00310                  IF (OPFCHK .GT. 1)   WRITE (6,INPUT1)                              000266
00314                  I1 = 1                                                             000276
00315         500      CONTINUE                                                           000301
00316         505      IF (NENG .EQ. 0  .OR.  NEXT .EQ. 0)   GO TO 510                    000301
00320                  READ (IFCTAP)  FTIME,FMACH,FALPHA,FBETA,DELE,   DELR,   FQPSI,     000310
00320                 1 ((THRUST(K,I), K=1,6), I=1,NUMENG),                               000310
00320                 2  ((XFORCE(K,I), K=1,6), I = 1,NUMXF),                             000310
00320                 3  (ACCEL(I), I=1,6), (CGLOC(I), I=1,3)                             000310
00357                  GO TO 540                                                          000356
00360         510      IF (NENG .EQ. 0)   GO TO 520                                       000360
00362                  READ (IFCTAP)  FTIME,FMACH,FALPHA,FBETA,DELE,   DELR,   FQPSI,     000361
00362                 1 ((THRUST(K,I), K=1,6), I=1,NUMENG),                               000361
00362                 2  (ACCEL(I), I=1,6), (CGLOC(I), I=1,3)                             000361
00412                  GO TO 540                                                          000416
00413         520      IF (NEXT .EQ. 0)   GO TO 530                                       000420
00415                  IF (EXTFOR)   GO TO 525                                            000421
00417                  READ (IFCTAP)  FTIME,FMACH,FALPHA,FBETA,DELE,   DELR,   FQPSI,     000430
00417         C       *  THRUST(1,I),                                                     000430
00417                 1  ((XFORCE(K,I), K=1,6), I = 1,NUMXF),                             000430
00417                 2  (ACCEL(I), I=1,6), (CGLOC(I), I=1,3)                             000430
00450                  GO TO 540                                                          000467
00451         525      READ (IFCTAP)  FTIME,FMACH,FALPHA,FBETA,DELE,   DELR,   FQPSI,     000471
00451                 1    (XFORCE(1,I), I = 1,NEXT),                                     000471
00451         C       *  THRUST(1,I),                                                     000471
00451                 2  (ACCEL(I), I=1,6), (CGLOC(I), I=1,3)                             000471
00477                  GO TO 540                                                          000531
00500         530      READ (IFCTAP)  FTIME,FMACH,FALPHA,FBETA,DELE,   DELR,   FQPSI,     000533
00500                 1  (ACCEL(I), I=1,6), (CGLOC(I), I=1,3)                             000533
00521         540      CONTINUE                                                           000560
00522                  IF ((FTIME - STARTM) .LE. STRTOF)    GO TO 505                     000560
00524                  TIME(I1) = FTIME                                                   000565
00525                  MACH(I1) = FMACH                                                   000570
00526                  ALPHA(I1) = FALPHA                                                 000572
00527                  BETA(I1) = FBETA                                                   000574
00530                  DELTAE(I1) = DELE                                                  000576
00531                  DELTAR(I1) = DELR                                                  000600
00532                  QPSI(I1) = FQPSI                                                   000602
00533                  IF (OPFCHK .GT. 1)   WRITE (6,INREPT)                              000604
00537                  IF (NLINK .EQ. 0  .OR.  NMODES .EQ. 0)   GO TO 610                 000621
```

```
     ******* FCREAD *******
                                                                            000631
00541              READ (IFCTAP) ((LNKFOR(K,I), K=1,3), I=1,NLNK),         000631
00541             1  (ZETA(K), K=1,NMODES),(ZETAD(K), K=1,NMODES), (ZETADD(K), 000631
00541             2  K=1,NMODES)                                          000664
00566              GO TO 630                                               000666
00567        610   IF (NLINK .EQ. 0)   GO TO 620                           000667
00571              READ (IFCTAP)  ((LNKFOR(K,I), K=1,3), I=1,NLNK)          000703
00602              GO TO 630                                               000705
00603        620   IF (NMODES .EQ. 0)   GOTO 630                           000706
00605              READ (IFCTAP)                                           000706
00605             1  (ZETA(K), K=1,NMODES),(ZETAD(K), K=1,NMODES), (ZETADD(K), 000706
00605             2  K=1,NMODES)                                          000735
00623        630   CONTINUE                                                000735
00624              IF (OPFCHK .GT. 2)   WRITE (6,LKMODS)                   000735
00624        C     REMOVE THIS AND THE NEXT CARD AFTR CHECKOUT.            000745
00630              IF (II .EQ. 40)   RETURN                                000754
00632              II = II + 1                                             000757
00633              IF ((FTIME - ENDTIM) .LT. ENDIFF)    GO TO 505          000765
00635              NUMINC = II - 1                                         000767
00636              RETURN                                                  001016
00637              END

     END OF COMPILATION:          NO  DIAGNOSTICS.

     BHD8v6  ******  FILE  ******
```

ORIGINAL PAGE IS
OF POOR QUALITY

4.4.3 Subroutine File

The purpose of subroutine FILE is to skip files on a tape.

STORAGE:

IBANK	40_8
DBANK	35_8

VARIABLES:

FNAME	I	1	Filename
NSKIP	I	1	Number of files to skip
ISTAT	I	1	Status variable for FSBSFL

ERROR MESSAGES:

- "IMPROPER STATUS VALUE FROM FSBSFL IS XXX.
  FILE NUMBER XXX CALLED BY XXXXXX."

For correction check the number of files on the referenced file to make sure there are as many as expected.

LISTING:

A listing of subroutine file is shown in Section 4.4.3.1

LIBRARY ROUTINES REQUIRED:

The FSBSFL routine is required by subroutine FILE to perform the file skip function. Appendix F has a description of this routine and its use.

```
                                                                                DATE 012375
****** FILE *****

?FOR,S P.FILE,FILE
FOR SE2C-01/23/75-00:30:53 (4,)

   SUBROUTINE FILE       ENTRY POINT 000031

   STORAGE USED: CODE(1) 000040; DATA(0) 000027; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    RNAME   000001

   EXTERNAL REFERENCES (BLOCK, NAME)

    0004    FSBSFL
    0005    NYDUS
    0006    NI02$
    0007    NERR3$

   STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000012 10L      0000   000001 1000F     0003 R 000000 CALING     0000   000023 INJP$     0000 I 000000

00101              SUBROUTINE FILE (FNAME, NSKIP)                              000000
00103              COMMON/RNAME/CALING                                         000000
00104              INTEGER   FNAME                                             000000
00105              CALL  FSBSFL (FNAME, NSKIP, ISTAT)                          000000
00106              IF (ISTAT .NE. 0)   GO TO 10                                000004
00110              RETURN                                                      000006
00111        10    WRITE (6,1000)   ISTAT, FNAME, CALING                       000012
00116        1000  FORMAT (///, 10X, 'IMPROPER STATUS VALUE FROM FSBSFL IS ', I3    000021
00116            1    ,5X, ' FILE NUMBER = ', I3, 4X, 'CALLED BY ',A6)         000021
00117              RETURN                                                      000021
00120              END                                                         000037
```

APPENDIX C

A Sample Document Written to Follow the Suggested Format

COMPUTER PROGRAM DOCUMENTATION

APPLIED/INERTIA LOADS TRANSFORM PROGRAM

(TRAIL)

by

F. M. Stratman

Prepared Under Project No. 3200 T

by

Lockheed Electronics Company, Inc.
Aerospace Systems Division

For

Structures and Mechanics Division

January 1975

National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

-48-

Applied Mechanics
Department 641-11
CPD 404

COMPUTER PROGRAM DOCUMENTATION

APPLIED/INERTIA LOADS TRANSFORM PROGRAM

(TRAIL)

Prepared by: [signature]

F. M. Stratman
Structures Technology Section

Approved by: [signature]

D. G. Probe, Supervisor
Structures Technology Section

[signature]

R. T. Theobald, Manager
Applied Mechanics Department

Lockheed Electronics Company, Inc.
Aerospace Systems Division
Houston, Texas

January 1975

LEC-3725

THIS FORM MUST BE COMPLETED BY TYPEWRITER

01 4	01 7 PROGRAM NO		01 14 DATE (MMDDYY)
JSC	G134	JSC COMPUTER PROGRAM ABSTRACT	5/22/74

01 20 TITLE OF PROGRAM (62 CHARACTERS MAXIMUM)	01 72 SYMBOLIC NAME (9 CHARACTERS MAXIMUM)	PARENT PROGRAM 02 14 CATEGORY	02 16 SITE	02 19 PROGRAM NO
Applied/Inertia Loads Transform Program (TRAIL)	TRAIL			

02 26 CATEGORY	02 27 LANGUAGE NO. 1	02 32 LANGUAGE NO. 2	02 37 KEY WORDS (8 MAXIMUM SEPARATED BY COMMAS)
G	FOR-5		Transformation, inertia loads, applied loads

WHOM TO CONTACT ABOUT THE PROGRAM

05 14 CONTACT (LAST NAME)	05 28 SITE	05 31 ORGN CODE	05 39 PROJECT NO	05 45 NASA CENTER
Schwartz	JSC	ES2	3200T	

05 48 STATUS
- ☐ A. UNDER DEVELOPMENT
- ☒ B. OPERATIONAL
- ☐ C. COMPLETED

05 49
- ☐ A. THIS PROGRAM IS NOT FOR SHARING
- ☒ B. LIMITED SHARING (SEE ABSTRACT)

DATES 05 50 INITIATED MMYY	05 54 COMPLETED MMYY	05 58 REVISION CODE	05 59 MAN-MONTHS	05 64 MACHINE HOURS	05 69 COMPUTER TYPE	05 74 TOTAL COST (DOLLARS)
0274	1074	☐ A. REVISION ☐ B. CANCELLATION	6.0	3.0	UNIVAC 1110	.

CARD NUMBER | COLUMN 8 | ABSTRACT | ELITE MARGIN | PICA MARGIN

This program transforms rigid and dynamic inertia loads and applied loads from the body loads model of the Integrated Structural Analysis System (ISAS) to the detailed finite element model. The program produces NASTRAN-Compatible output.

RELATED DOCUMENTATION (66 CHARACTERS MAXIMUM SEPARATE EACH REF BY COMMAS)

JSC Form 142 (Aug 73) NASA-JSC

TABLE OF CONTENTS

		Page
1.0	INTRODUCTION	1-1
2.0	PROGRAM FUNCTIONAL DESCRIPTION	2-1
2.1	General Description	2-1
2.1.1	Rigid Body Inertia Forces	2-1
2.1.2	Modal Inertia Forces	2-1
2.1.3	Transformation to the Finite Element Model	2-3
2.2	Procedure	2-4
3.0	USAGE	3-1
3.1	Program Organization	3-1
3.1.1	Operating Characteristics	3-1
3.2	Input Description	3-2
3.2.1	Control Deck Setup	3-3
3.2.2	Data Deck Setup	3-4
3.2.3	Description of Input Files	3-6
3.2.3.1	User Modal File	3-6
3.2.3.2	User Weights File	3-7
3.2.3.2.1	CONM1 Description	3-8
3.2.3.2.2	GRID Description	3-9
3.2.3.3	Compressed Aerodynamics Forces and Conditions File	3-11
3.2.3.4	Batch Flight Conditions File	3-16
3.2.3.5	Transformation Matrix File	3-17
3.3	Output Description	3-18
3.3.1	Output Echo of Input	3-18
3.3.2	Data Output	3-18
3.3.2.1	NASTRAN Format Punched Cards	3-19
3.3.2.1.1	FORCE Card Description	3-19
3.3.2.1.2	MOMENT Card Description	3-20

-51-

TABLE OF CONTENTS (cont'd)

	Page
3.3.2.2 NASTRAN Input Loads File (NASFIL) Format	3-22
3.3.3 Diagnostic Output	3-25
3.4 Sample Problem	3-26
3.4.1 Sample Problem Description	3-26
3.4.2 Sample Problem Input	3-28
3.4.2.1 Sample Run Request (588)	3-29
3.4.2.2 Sample Deck Setup	3-30
3.4.2.3 Sample Input Deck	3-31
3.4.2.4 Sample Problem User Modal File Input	3-33
3.4.2.5 Sample Problem User Weights File Input	3-34
3.4.3 Sample Problem Output	3-35
3.4.3.1 Problem #1 Output Listing . . .	3-35
3.4.3.2 Problem #2 Output Listing . . .	3-43
3.4.3.3 Problem #3 Output Listing . . .	3-50
4.0 SUBROUTINE DOCUMENTATION	4-1
5.0 REFERENCES	5-1

-52-

1.0 INTRODUCTION

TRAIL is an ISAS program intended for use as a batch run program. Its purpose is to transform body loads from the minimum node body load model to the finite element model which has an expanded number of nodes.

This program calculates forces and moments for the finite element model. The types of forces and moments which are considered by this program include modal inertia, rigid body inertia, and applied external forces and moments.

Loads Model input data comes from several files. Rigid body inertia data can be obtained from the User Modal File or User Weights File. Modal data is input by the User Modal File. Applied Forces and Moments are copied from the Compressed Aerodynamic Forces and Conditions File. Inertial and Modal accelerations are copied from the Flight Conditions File. Transformation equations are set up for input by the analyst. For checkout of this program, transformation matrices were input by cards. The intent for future ISAS work is to have often-used transformation matrices stored in the Transform Equations File.

Program output is in the form of a NASTRAN compatible file of forces and moments. The output is formatted as NASTRAN Force and Moment cards and stored in the NASTRAN Input Loads File. NASTRAN bulk-data, FORCE and MOMENT cards can also be output.

Requirements for this program and its documentation were obtained from References 1, 2, and 3.

2.0 PROGRAM FUNCTIONAL DESCRIPTION

The computations performed by this program include the determination of modal acceleration forces and the rigid body inertia forces for each node-degree-of-freedom. When these forces are calculated for the loads model and after external loads are read into the program the forces are transformed to the finite element model. The flow and mechanics of the calculations performed by this program are shown in subsections of part 2.0 of this document.

2.1 General Description

The basic flow of the program is under the control of routine MAIN and is illustrated by the flow chart in Figure 2-1.

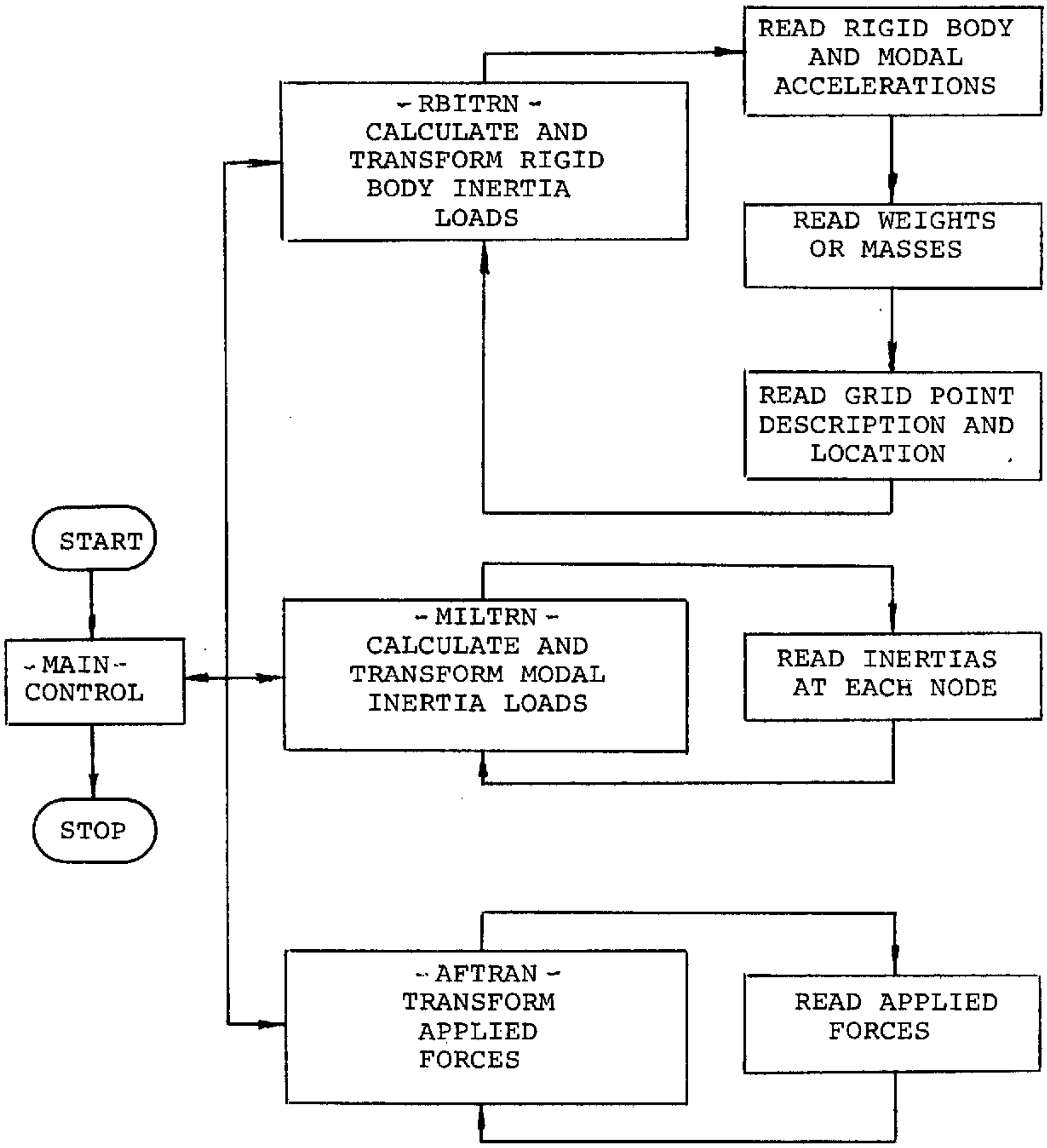
2.1.1 Rigid Body Inertia Forces

The matrix of rigid-body forces and moments {FRB} acting on the loads model is calculated by postmultiplying the inertia matrix, GTM, by the six accelerations about the total vehicle center-of-gravity. GTM is an NDF X 6 matrix which is obtained by premultiplying [G], the location vector matrix by the [MASS] matrix.

$$[GTM] = [MASS]\ [G]$$
$$[FRB] = [GTM]\ [-ACCEL]$$

2.1.2 Modal Inertia Forces

The modal matrix, PHI from the User Modal File (UMF) is postmultiplied by the matrix of modal accelerations $[\ddot{Q}]$ to obtain the matrix of nodal accelerations, {NODACC}. {NODACC} postmultiplies the MASS matrix from the User Modal File, creating the array of modal forces and moments, [FMOD]



<u>BASIC FLOW OF THE TRAIL PROGRAM</u>

Figure 2-1

-55-

$$\{NODACC\} = [PHI]\ \{\ddot{Q}\}$$

$$\{FMOD\} = [-MASS]\ \{NODACC\}\ .$$

2.1.3 Transformation to the Finite Element Model

The matrix multiplications used to transform body loads forces to the finite element model are illustrated in Figure 2-2. The FFE matrices represent the finite element loads.

rigid body inertia: $$\underset{NDF_{FE}}{\{FFE1\}} = [TRB]\ \underset{NDF}{\{FRB\}}$$

Applied: $$\underset{NDF_{FE}}{\{FFE2\}} = [TAP]\ \underset{NDF}{\{FAP\}}$$

Modal: $$\underset{NDF_{FE}}{\{FFE3\}} \quad [TMOD]\ \underset{NDF}{\{FMOD\}}$$

TRANSFORMATION EQUATIONS

Figure 2-2

The "T" matrices are transform matrices set up by the user. FRB, FAP, and FMOD are the forces and moments acting on the loads model, NDF is the number of degrees-of-freedom of the loads model and NDF_{FE} is the number of degrees-of-freedom of the finite element model.

-56-

2.2 Procedure

The FRB, FMOD, and FAP matrices are set up in subroutines RBITRN, MILTRN, and AFTRAN respectively. The documentation of these routines describe the processes used to set-up the matrices. All three routines use subroutine TRANS to do the actual transform. NASTRAN bulk data cards are punched for each computed finite element force or moment. The NASTRAN card data is also used to create card images which are buffered into 20-card blocks and written on the NASTRAN INPUT Loads File using NTRAN.

-57-

3.0 USAGE

The TRAIL Program was written in Fortran V and was developed on a Univac 1110 computer with an EXEC 8 system to be utilized within the ISAS. There are five ISAS files which are used by TRAIL as input data. The output can be either in the form of punched force and moment cards in NASTRAN format or card images on a NASTRAN Input Loads File.

Program control is setup by information read into several namelist groups. There is some namelist information which has been incorporated into the program for checkout purposes and will provide an optional means of data input for batch run operation. The namelist can be initialized by reading cards or, when ISAS is operable, from terminal input. The transformation matrices are presently read into TRAIL as namelist data because there was no ISAS file available at the time of the development of the TRAIL Program. This was also the case with the User Weights File and the Compressed Aerodynamic Forces and Conditions File.

3.1 Program Organization

A general flow chart of the program is presented in Figure 2-1.

3.1.1 Operating Characteristics

Some of the problem size limitations are described below.

Maximum size (Finite Element model)	-	degrees-of-freedom	= 410
		number of modes	= 100
		number of nodes	= 120

The above dimensions result in a total D and I bank usage of 57 K

number of cases - one

A checkout of a problem with a 27 degree-of-freedom finite element model used less than one minute of computer time and thirty pages of print were output. Since no higher degree-of-freedom checkout problems were run. An accurate estimate of time and pages cannot be made. The program user should observe his program run time as he is getting familiar with the program to obtain accurate estimates for future computer run submission.

3.2 Input Description

Input for this program is composed of a control deck, a data deck, and a series of input files. The description of the control deck is shown in Figure 3-1 and details of input file data and namelist follow.

-59-

3.2.1 Control Deck Setup

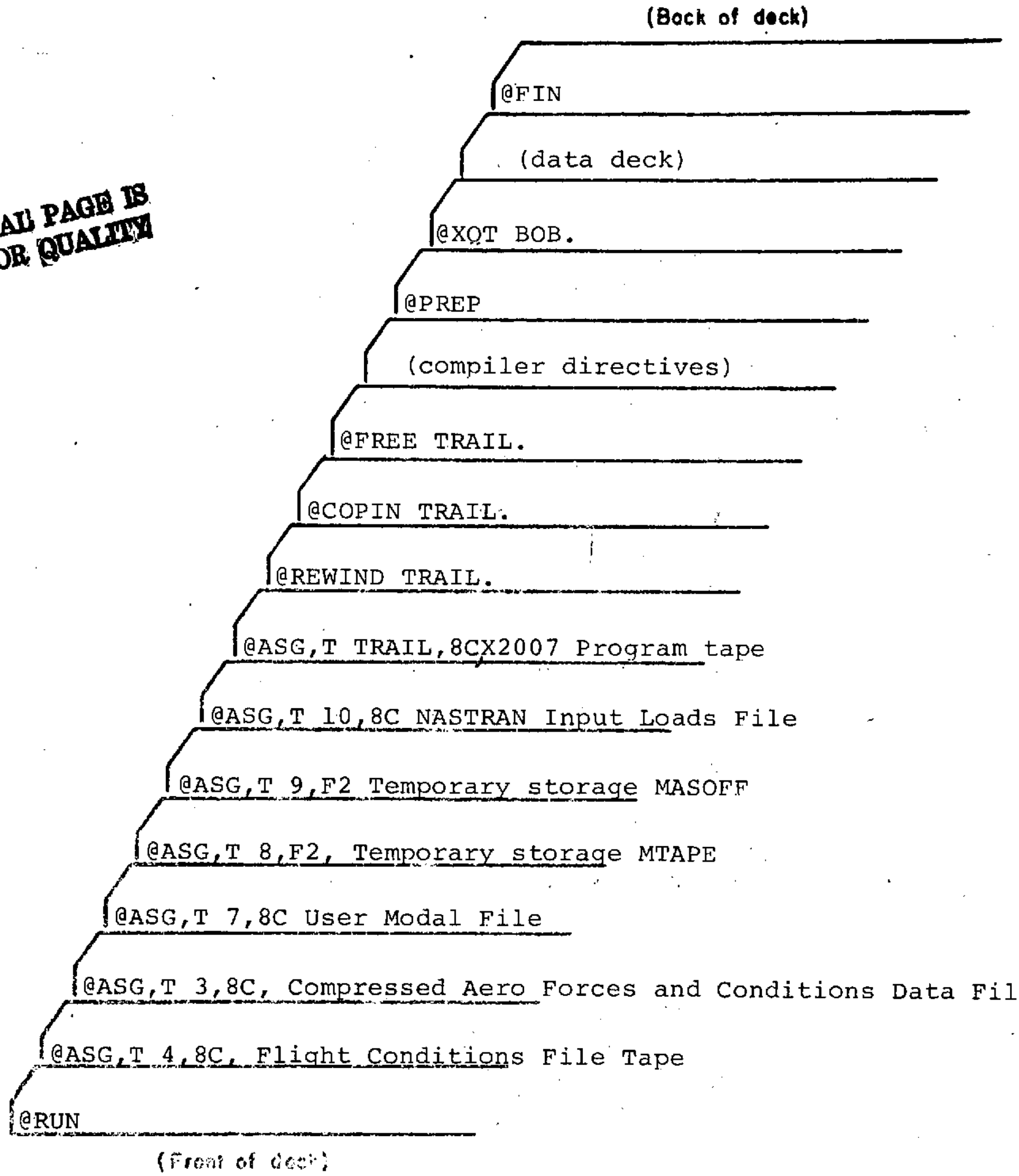


FIGURE 3-1 DECK SETUP

-60-

3.2.2 Data Deck Setup

$TPDAT	= Namelist Input
CMASS	= Scale factor to convert mass to weight, default is 1.0
CINERT	= Scale factor to convert mass portion of the moment-of-inertia to weight, default is 1.0
SID	= Load set i.d. for creating NASTRAN card images
XCG,YCG,ZCG	=cg. of the entire body
UMF	= .TRUE., read mass data from User Modal File .FALSE., read mass data from User Weights File
OFFDIG	= .TRUE., off diagonal elements in the mass matrix
FFREAD	= .TRUE., read flight conditions tape .FALSE., flight conditions file not used
TIME	= Time of desired flight condition
CARDS	= If .FALSE. punching of FORCE and MOMENT cards suppressed-default is .TRUE.
NUMF	= Position of modal data on User Modal File
MODINR RIGID APPLYD	= Modal inertia, rigid body inertia, and applied loads computation flags, respectively - if .FALSE. the corresponding load calculations are suppressed.
TRNOUT MASOUT MODLOT	= Transform array, mass array, and modal data output echo flags respectively - if .FALSE. the output of corresponding array is suppressed. Default is .TRUE.

3.2.2 Data Deck Setup (continued)

ACCEL = 6 accelerations read in if flight conditions file not used. Units = in/sec^2, rad./sec^2.

$END

NNODES I3 Format - Number of Nodes

CONM1 cards

@EOF

GRID cards

@EOF

(These cards are used only if the User Weights File is used. (UMF=.FALSE.))

$ASSGN - namelist input

DATNAM = 6HTRBbbb, identifies following data as rigid body transform arrays

NDFFE No. of d.o.f. of finite element model

ASSGN1 (I,J), J=1,3), I = 1, NDFFE Provides information about finite element data being computed See subroutine documentation of TRANS

$END Namelist Input

repeated for each row of TRB:

$ARR

IROW row number of transform array

TRB values corresponding to IROW

$END

@EOF

$ASSGN - Namelist Input

DATNAM = 6HTMODbb, identifies following arrays as modal transform data optional if identical to value used for rigid body transofrm

ASSGN2 (I,J), J=1,3), I=1, NDFFE Provides information about finite element data being created - see subroutine documentation of TRANS

$END - Namelist Input

-62-

3.2.2 Data Deck Setup (continued)

```
                  ( $ARR
repeated          | IROW        row number of transform array
for each          | TMOD        values corresponding to IROW
row of            ( $END
TMOD                @EOF
$ASSGN              Namelist Input
DATNAM            = 6HTAPbbb,   identifies following arrays as
                                applied force transform data
NDFFE                           optional if identical to value used
                                for preceding portion
ASSGN3(I,J), J=1,3), I=1, NDFFE - Provides information about
                    finite element data being created - See
                    subroutine documentation of TRANS
$END                Namelist input

                  ( $ARR
repeated          | IROW        row number of TAP array
for each          | TAP         values corresponding to IROW
row of            ( $END
TAP                 @EOF
```

An example of input deck data can be seen in the sample problem input deck. See Section 3.4.2.3.

3.2.3 Description of Input Files

The description of each ISAS input file follows. The formats indicated were obtained from the ISAS Functional Specifications Document, Reference 2.

3.2.3.1 User Modal File (Description of one individual file).

Created by: NASTOF

Output Method: Binary

Record 1 - 12 word title

Record 2 - NDF, NMODES, (IDXYZ(I), I=1,NDF), NNODES, (NODEID(J),J=1,NNODES), (XYZ(I,J),J=1,3; I=1,NNODES),(DMASS(K),K=1,NDF),GMASS

63-

Record 3

FREQ,(PHI(J),J=1,NDF)

Record NMODES+2

Record NMODES+3 NODEID(I),(NAME(K,I),K=1,4),
I=1,NNODES)

Record NMODES+4 (RMASS(I),I=1,NDF)

Record NMODES+3+NDF

Note: The last two items may not exist on all tapes.

3.2.3.2 User Weights File

This file does not exist at the present time, however this program uses the NASTRAN bulk data cards CONM1 and GRID to simulate the User Weights File. The format of these cards is shown in Section 3.2.3.2.1 and 3.2.3.2.2 respectively.

3.2.3.2.1 Bulk Data Deck - CONM1

Input Data Card CONM1 Concentrated Mass Element Connection

Description: Defines 6X6 symmetric mass matrix at a geometric grid point of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CØNM1	EID	G	CID	M11	M21	M22	M31	M32	abc
CØNM1	2	22	2	2.9		6.3			+1
+bc	M33	M41	M42	M43	M44	M51	M52	M53	def
+1	4.8				28.6				+2
+ef	M54	M55	M61	M62	M63	M64	M65	M66	
+2		28.6						28.6	

Field	Contents
EID	Unique element identification number (Integer > 0)
G	Grid point identification number (Integer > 0)
CID	Coordinate system identification number for the mass matrix (Integer ≥ 0)
Mij	Mass matrix values (Real)

Remarks: 1. Element identification numbers must be unique with respect to all other element identification numbers.

-65-

3.2.3.2.2 Bulk Data Deck-GRID

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		316		

Field	Contents
ID	Grid point identification number (Integer > 0)
CP	Identification number of coordinate system in which the location of the grid point is defined (Integer ≥ 0 or blank*).
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 (Integer ≥ 0 or blank*)
PS	Permanent single-point constraints associated with grid point (any of the digits 1-6 with no imbedded blanks) (Integer ≥ 0 or blank*)

Remarks: 1. All grid point identification numbers must be unique with respect to all other structural, scalar, and fluid points in each NASTRAN run.

2. The meaning of X1, X2 and X3 depend on the type of coordinate system, CP, as follows: (see CORD__ card descriptions)

-66-

Type	X1	X2	X3
Rectangular	X	Y	Z
Cylindrical	R	Θ (degrees)	Z
Spherical	R	Θ (degrees)	Φ (degrees)

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.

* See the GRDSET card in Reference 4 for default options for fields 3, 7 and 8.

Additional details on the use of these card formats can be found in Reference 4, The Nastran Users Manual.

-67-

3.2.3.3 Compressed Aerodynamic Forces and Conditions File format.

56 words — Header Information Block (cont.)

	Description
24 words	Title information
1 word	Logical variable (ENGID) to identify engines True – use engine ID numbers False – use X,Y,Z coordinates
1 word	Logical variable (EXTFØR) to identify external forces True – use ID numbers False – use X,Y,Z coordinates
1 word	Logical variable (LNK) to identify link forces True – use ID numbers False – use X,Y,Z coordinates
1 word	Number of engines = NENG (max. = 30)
1 word	Number of external forces = NEXT (max. = 200)
1 word	Number of link forces = NLINK (max. = 50)
1 word	Number of modes = NMØDES (max. = 100)
1 word	Start time for data contained on this file
1 word	Stop time for data contained on this file
1 word	Number of times contained on this file = NØTIME
1 word	Number of points = NØGP (max. = 200)
1 word	Beginning word location of data blocks

		Description
Header Information Block (conc.)	1 word	Number of words per data block
	1 word	= 0 – link forces of Flight Conditions File are used = 1 – link forces are to be computed
	18 words	Spare
Engine Identification Block	NENG or (NENG/6)3	Logical variable (ENGID) to identify engines True – use engine ID numbers False – use X,Y,Z coordinates
External Force Identification Block	NEXT or (NEXT/6)3	Logical variable (EXTFØR) to identify external forces True – use ID numbers False – use X,Y,Z coordinates
Mode Identification Block	NMØDES	Identification numbers for each of the modes
Link Identification Block	NLINK or (NLINK/6)3	Logical variable (LNK) to identify link forces True – use ID numbers False – use X,Y,Z coordinates
Point ID Block	NØGP	Point identification numbers
Point Coordinate Block	NØGP*3	X,Y,Z coordinates of each point
Data Block for First Time (cont.)	1 word	Time
	1 word	Mach number
	1 word	Angle of attack
	1 word	Angle of sideslip

69

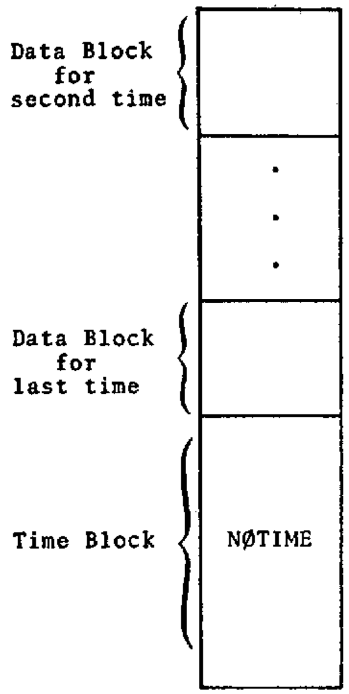
	Size	Description
(16 + NENG + NEXT + NLINK + 3*NMØDES + 6*NØGP) words (cont.) Data Block for first time (cont.)	1 word	Elevon deflection
	1 word	Dynamic pressure
	3 words	X,Y,Z coordinates of center of gravity
	3 words	X,Y,Z directions translational rigid body acceleration
	3 words	X,Y,Z directions rotational rigid body acceleration
	NENG	Thrust for engine number 1 ⋮ Thrust for engine number NENG
	NEXT	Force for external force number 1 ⋮ Force for external force number NEXT
	NMØDES	Modal acceleration for first mode ⋮ Modal acceleration for last mode

(16 + NENG + NEXT + NLINK + 3*NMØDES + 6*NØGP words (conc.)

Data Block for first time (conc.)

	Description
NMØDES	Modal velocity for first mode . . . Modal velocity for last mode
NMØDES	Modal displacement for first mode . . . Modal displacement for last mode
NLINK	Force for first link . . . Force for last link
3 words	X,Y,Z Force coefficient components for first load station
3 words	X,Y,Z Moment coefficient components for first load station
. . .	. . .
3 words	X,Y,Z Force coefficient components for last load station
3 words	X,Y,Z Moment coefficient components for last load station

Description



Time for first data block on this file

.
.
.

Time for last data block on this file

-72-

3.2.3.4 BATCH Flight Conditions File Format

Record 1 - 24 word title with the first two words identifying the program that generated the data.

Record 2 -

ENGID	-	Logical
TRUE	→	Engine I.D. numbers used
FALSE	→	Engine XYZ locations used
EXTFOR	-	Logical
TRUE	→	External Force I.D. numbers
FALSE	→	External Force XYZ locations used
NENG	-	Number of engine thrust components, = 6 per external force if ENGID false
NEXT	-	Number of external forces and moments = 6 per external force point if EXTFOR false
NLINK	-	Number of link forces and moments, = 6 per link if LNK false
NMODES	-	Number of modes
LNK	-	Logical
TRUE	→	Link I.D. numbers used
FALSE	→	Link XYZ locations used
START	-	Start time of simulation
STOP	-	Stop time of simulation

Record 3 - (If NENG = 0, No Record 3.)
If ENGID true, IDENG(I), I=1, NENG
If ENGID false, (ENGXYZ(K,I),K=1,3),I=1,NENG/6

Record 4 - (If NEXT = 0, No Record 4.)
If EXTFOR true, IDEXT(I),I=1,NEXT
If EXTFOR false (EXTXYZ(K,I),K=1,3),I=1,NEXT/6

Record 5 - If NMODES >0 MODES(I),I=1,NMODES
If NMODES <0 no Record 5.

Record 6 - (If NLINK = 0, no Record 6.)
If LNK true, IDLINK(I),I=1,NLINK
If LNK false, (XYZLNK(K,I),K=1,3),I=1,NLINK/6

Record 7A - TIME, Mach number, angle of attack, yaw angle, elevon deflection, rudder deflection, dynamic pressure, engine thrusts*, non-aerodynamic external forces*, C. G. Rigid-body accelerations (linear and angular), c.g. locations

Record 7B - has 4 alternatives

i) NLINK=0 and NMODES=0 - no Record 7B.
ii) NLINK=0 and NMODES>0 - modal accelerations, velocities and displacement
iii) NLINK 0 and NMODES=0 - Link Forces
iv) NLINK 0 and NMODES>0 - Link Forces and modal accelerations, velocities and displacements

Record 7 is repeated for each time step between START and STOP

*Optional data - If number of engines is zero (NENG=0) or number of external forces is zero (NEXT=0) their respective thrust and forces will not appear in the list.

3.2.3.5 Transformation Matrix File

Details to be supplied at a later date.

3.3 Output Description

This program as developed has the option of producing punched card output or a data file containing card images. The format of each type output is intended to be used as input to NASTRAN. The format of both modes of output is therefore NASTRAN-compatible.

3.3.1 Output Echo of Input

The following input items are echoed as printout:

1. Loads Model Force Matrices
2. Transform matrix row by row(Suppressed if TRNOUT = .FALSE.)
3. Rigid body and modal acceleration
4. Number of degrees of freedom
5. Number of modes (Loads Model)
6. Number of nodes (Loads Model)
7. Transform parameters
8. Node i.d.'s
9. Element i.d.'s
10. Mass matrices (Suppressed if MASOUT = .FALSE.)
11. Modal data (Suppressed if MODLOT = .FALSE.)

Section 3.4.3 illustrates the echoed input data.

3.3.2 Data Output

A printout of the NASTRAN FORCE or MOMENT card corresponding to each non-zero row of the transform matrix is printed out. In the listing it follows the output echo of the non-zero row. Section 3.4.3 illustrates the data output.

3.3.2.1 NASTRAN Format Punched Cards

A NASTRAN FORCE or MOMENT card is also punched by default. The operation can be suppressed by inputting a value of .FALSE. for logical variable, CARDS, in NAMELIST, TPDAT.

The descriptions of the NASTRAN Format bulk data cards follows.

3.3.2.1.1 Bulk Data Deck

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
FØRCE	SID	G	CID	F	N1	N2	N3		
FØRCE	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 $\geq$ 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

$$\vec{f} = - \vec{N}$$

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

-76-

Field	Contents
SID	Load set identification number (Integer >0)
G	Grid point identification number (Integer >0)
CID	Coordinate system identification number (Integer $\geq$ 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

$$\vec{f} = - \vec{N}$$

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.

3.3.2.1.2 Bulk Data Deck - MOMENT

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
MØMENT	SID	G	CID	M	N1	N2	N3		
MØMENT	2	5	6	2.9	0.0	1.0	0.0		

-77-

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

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

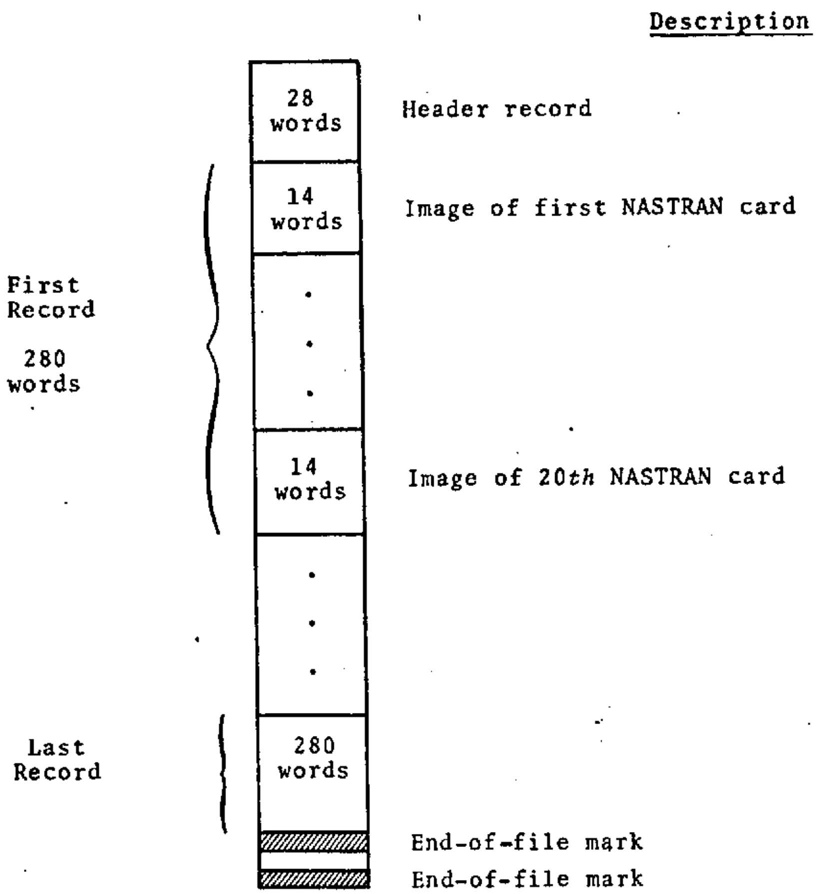
2. Load sets must be selected in the Case Control Deck (LOAD=SID) to be used by NASTRAN.

3. A CID of zero references the basic coordinate system.

-78-

3.3.2.2 NASTRAN Input Loads File format.

(NASFIL)



Header Record format:

Words	Type	Description
20 words	A	120 characters of file identification information
1 word	I	Number of NASTRAN card images
7 words		Spare

-80-

Detail format of a NASTRAN card image: Fieldata characters for the 14 words on a NASTRAN card are contained in the card columns shown below.

Word	Card Column
1	1-6
2	7-12
3	13-18
4	19-24
5	25-30
6	31-36
7	37-42
8	43-48
9	49-54
10	55-60
11	61-66
12	67-72
13	73-78
14	79-80

3.3.3 Diagnostic Output

When reading the mass matrices from the USER WEIGHTS FILE in routine RBITRN, if data cards are out of sequence the following message is printed:

DATA CARD OUT OF SEQUENCE NEAR ELEMENT ID NO. ____________

The type of data being transformed is passed as a Hollerith variable to the argument list of subroutine TRANS by the subroutine which sets up the transformation. This variable is compared to the title of the transform matrix deck and if they are not equal the following diagnostic is printed and execution is terminated.

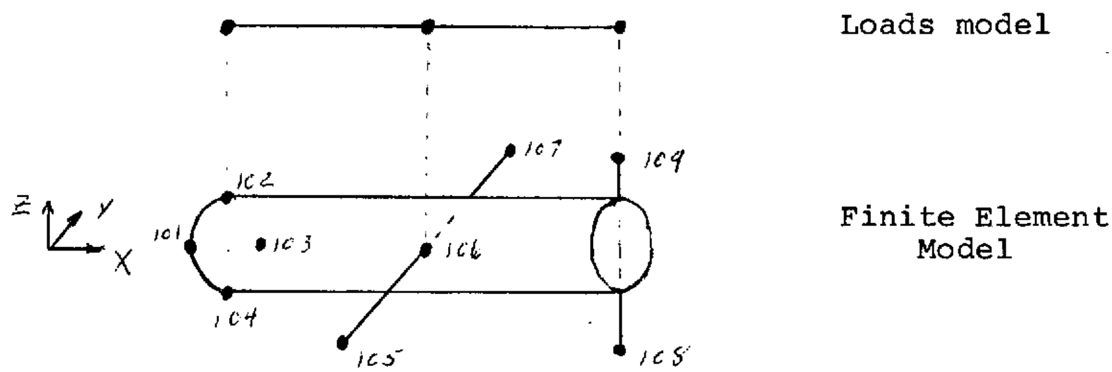
DATA DECK OUT OF SEQUENCE

3.4 Sample Problem

A simple test problem was used to checkout the programs integrity. A small loads model consisting of three concentrated masses on a simple beam was used as the input case. The program was expected to transform the loads supplied, which were referenced to this three node model, to a finite element model with nine node points and twenty-seven degrees-of-freedom.

3.4.1 Sample Problem Description

The loads model and the expanded finite element model are illustrated in Figure 3-2. The model is a rough representation of an aircraft structure.



$NDF = 9$

$NDF_{FE} = 27$

FIGURE 3-2

LOADS MODEL AND FINITE ELEMENT MODEL USED FOR SAMPLE PROBLEM

To obtain a set of Flight Conditions which the structure could be subjected to, the six components of rigid-body acceleration and the modal accelerations were copied from a Flight Conditions File which was produced by a Space Shuttle Flight Simulation Program.

The program transforms all three types of loads in one run but three sample problems were run to checkout the numerous options the program has for input. For the first sample problem mass property data was obtained from a simulated User Weights File. Inertial loads were transformed by this check case. Both mass and modal properties were obtained from a User Modal File for the second run. Only rigid body and modal inertia loads were transformed in this sample. The last check case was run using a DATA statement in subroutine AFTRAN to provide externally applied force information so a transform of external forces could be checked. This procedure was followed because there was no Compressed Aerodynamic Forces and Conditions File available. The ISAS File, Compressed Aerodynamic Forces and Conditions, which will be the source of input applied loads is not presently available. The following FAP array was placed in a data statement in Subroutine AFTRAN.

$$\begin{Bmatrix} 0 \\ 0 \\ 200. \\ 0 \\ -100. \\ 1000. \\ 100. \\ 0. \\ 0. \end{Bmatrix}$$

3.4.2 Sample Problem Input

The sample problem input includes a sample 588 for run number 2 as well as a sample deck setup for the same run. A sample run listing is shown in section 3.4.2.3 for problem #2.

The sample listing does not include the transformation data for transforming modal loads (TMOD). However, this data is similar to the $ASSGN and $ARR NAMELIST input for the rigid body transform.

3.4.2.1 Sample Run Request

INSTRUCTIONS FOR CENTRAL COMPUTER COMPLEX COMPUTER RUNS

(DO NOT FILL IN SHADED AREAS)

PROGRAMMER'S COMMENTS:

Punched Card Output expected — EXEC 8 1110

FSAG2

PROGRAMMER	BADGE NO	BOX NO.	PHONE NO.	DATE	PRIORITY & INITIALS
STRATMAN	L78162	AG2	X340	5/15	

DIVISION CODE	PROG. NO.	PROJ. NO.	EST. TIME	MAX. TIME	PAGES OUTPUT	SEG. NO.
EX2		3200	2	2	100	

OPERATING SYSTEM		TYPE OF RUN	NO. TAPES	NO. FASTR FILES	NO. DRUM FILES
1108 EXEC II [ ]	3200 SCOPE [ ]	PROD. [ ] TEST [X]	4	2	
1108 EXEC VIII [X]	3200 SMARTS [ ]	OTHER (EXPLAIN BELOW)			
1108 COBOL [ ]	3200 OTHER [ ]				

INPUT TAPES UNIT	REEL NO.	FILE NAME	WORKING TAPES	OUTPUT TAPES UNIT	REEL NO.	FILE NAME	SAVE	PERMANENT FASTRAND FILES
	X2007	TRAIL		10	X324		X	S
7	X194							S
4	X1269							S
								S
								S
								S

4060 [ ] 16 MM [ ] 35 MM [ ]	REEL NO.	FILE NO.	PUNCHED OUTPUT [X]	REEL NO.	NO. CARDS
					81

CAL COMP PLOT [ ]		NO. PLOTS	ACTUAL TIME USAGE

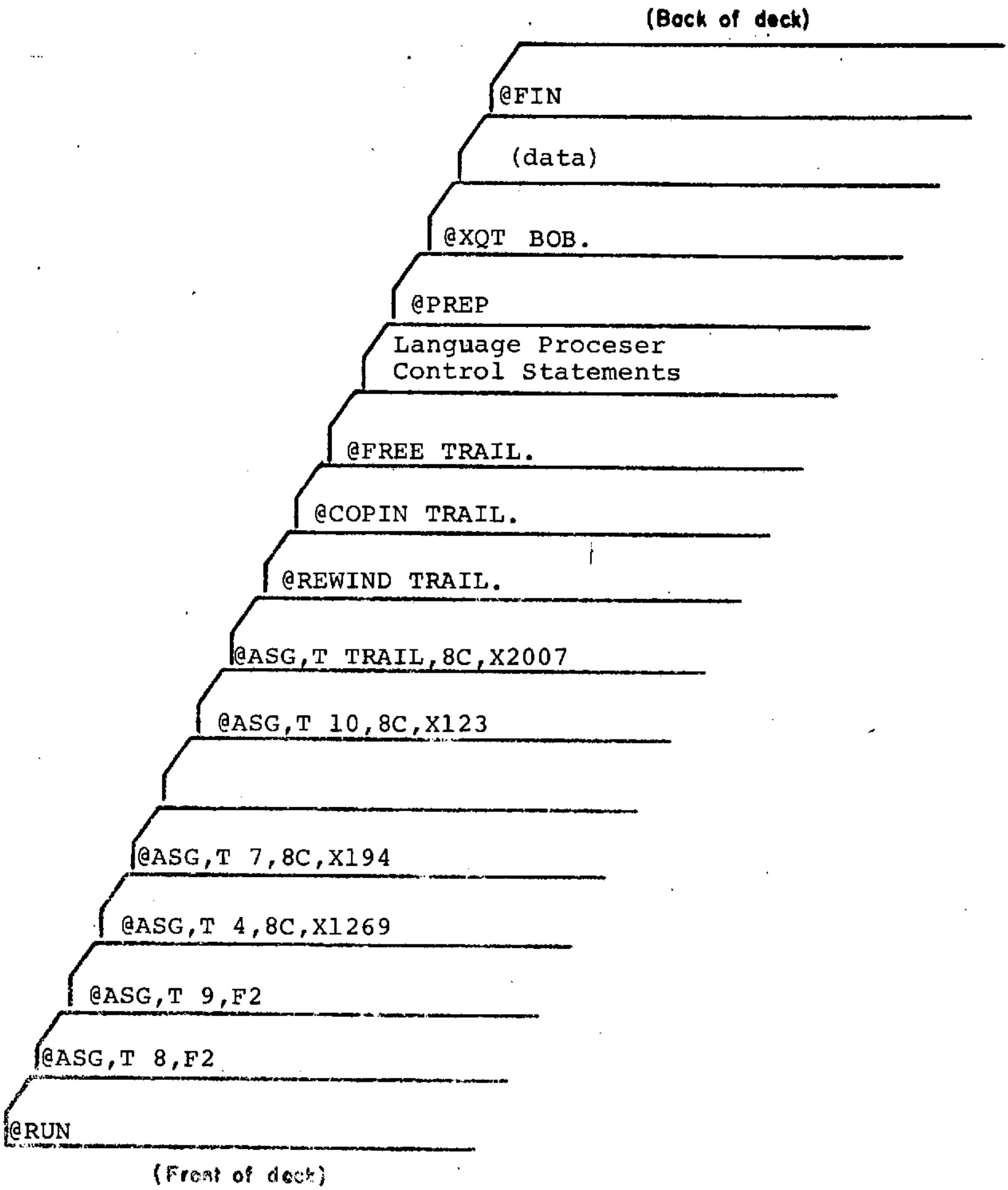
ABNORMAL STOPS		SYSTEM NO.
EXCESS OUTPUT [ ]		
EXCESS TIME [ ]		PAGES OUTPUT
OTHER (EXPLAIN BELOW) [ ]		

START

ORIGINAL PAGE IS OF POOR QUALITY

-86-

3.4.2.2 Sample Deck Setup



SAMPLE DECK SETUP FOR PROBLEM #2

3.4.2.3 SAMPLE INPUT DECK

```
RUN,/RT     FSH4,3200-G066-C,ES2-L78162, 1,100                    FRED STRAT
ASG,T    8,F2
ASG,T    9,F2
ASG,T    10,8C
ASG,T    4,8C,X1269
ASG,T    7,8C,X194
ASG,T    TRAIL,8C,X3194
REWIND   TRAIL.
COPIN    TRAIL.
FREE     TRAIL.
FOR,S    MAIN,MAIN
FOR,S    MILTRN,MILTRN
FOR,S    AFTRAN,AFTRAN
FOR,S    RBITRN,RBITRN
FOR,S    RDUMF,RDUMF
FOR,S    TRANS,TRANS
PREP
PRT,T
XQT  808
$TPDAT
APPLYD = .FALSE.
NUMF = 1
FFREAD = .TRUE.
TIME = 30.
XCG=2.,YCG=0.,ZCG=0.,
OFFDIG = .TRUE.
   UMF = .TRUE.
CARDS = .FALSE.,
$END
$ASSGN
        DATNAM=6HTRB    , NDFFE=27
ASSGN1(1,1)=101,101,101,102,102,102,103,103,103,104,104,104,105,105,
105,106,106,106,107,107,107,108,108,108,109,109,109,
ASSGN1(1,2)=27*1,ASSGN1(1,3)=1,2,3,1,2,3,1,2,3,1,2,3,1,2,3,1,2,3,
1,2,3,1,2,3,1,2,3,
$END
$ARR
IROW=1, TRB(1)=.25
$END
$ARR
IROW=2, TRB(2)=.25
$END
$ARR
IROW=3, TRB(3)=.25
$END
$ARR
IROW=4, TRB(1)=.25
$END
$ARR
IROW=5, TRB(2)=.25
$END
$ARR
IROW=6, TRB(3)=.25
$END
$ARR
IROW=7, TRB(1)=.25
$END
$ARR
IROW=8, TRB(2)=.25
$END
```

-88-

```
$ARR
IROW=9, TRB(3)=.25,
$END
$ARR
IROW=10, TRB(1 )=.25,
$END
$ARR
IROW=11, TRB(2 )=.25,
$END
$ARR
IROW=12, TRB(3 )=.25,
$END
$ARR
IROW=13, TRB(4 )=.333,
$END
$ARR
IROW=14, TRB(5 )=.333,
$END
$ARR
IROW=15, TRB(6 )=.333,
$END
$ARR
IROW=16, TRB(4 )=.333,
$END
$ARR
IROW=17, TRB(5 )=.333,
$END
$ARR
IROW=18, TRB(6 )=.333,
$END
$ARR
IROW=19, TRB(4 )=.333,
$END
$ARR
IROW=22, TRB(7 )=.5,
$END
$ARR
IROW=23, TRB(8 )=.5,
$END
$ARR
IROW=24, TRB(9 )=.5,
$END
$ARR
IROW=25, TRB(7 )=.5,
$END
$ARR
IROW=26, TRB(8 )=.5
$END
$ARR
IROW=27, TRB(9 )=.5
$END
EOF
```

-89-

3.4.2.4 Sample Problem User Modal File Input

The User Modal File contains grid point identification and location information, mass and inertia data at each grid point, and modal frequencies and shapes. The format of this input file can be seen in Section 3.2.3.1. The input data on the User Modal File which is used by problem #2 is printed out on the sample output listing.

3.4.2.5 Sample Problem User Weights File Input

The input deck for the simulated User Weights File is reproduced below. The transformation data portion is not reproduced on the listing. However, it is identical to the rigid body transformation data illustrated in Section 3.4.2.3.

```
RUN,/RT     FSH4,3200-G066-C,ES2-L78162, 1,100                FRED STRATMAN
ASG,T    8,F2
ASG,T    9,F2
ASG,T   10,F2
ASG,T    4,8C,X1269
ASG,T    TRAIL,8C,X2007
REWIND  TRAIL.
COPIN   TRAIL.
FREE    TRAIL.
FOR,S    AFTRAN,AFTRAN
FOR,S    RBITRN,RBITRN
FOR,S    TRANS,TRANS
PREP
XQT    BOB
$IPDAT
APPLYD = .FALSE.
MODINR = .FALSE.
CARDS  = .FALSE.
FFREAD = .TRUE.
TIME = 30.
XCG=2.,YCG=0.,ZCG=0.,
OFFDIG = .TRUE.
   UMF = .FALSE.
$END
 6
CONM1         102        2        02.0     1.0     2.0     1.0     1.0          1
       12.0     1.0     1.0      1.0     2.0     1.0     1.0     1.0          2
       21.0     2.0     1.0      1.0     1.0     1.0     1.0     2.0
CONM1         103        3        02.0     1.0     2.0     1.0     1.0          4
       42.0     1.0     1.0      1.0     2.0     1.0     1.0     1.0          5
       51.0     2.0     1.0      1.0     1.0     1.0     1.0     2.0
CONM1         104        4        02.0     1.0     2.0     1.0     1.0          7
       72.0     1.0     1.0      1.0     2.0     1.0     1.0     1.0          8
       81.0     2.0     1.0      1.0     1.0     1.0     1.0     2.0
 EOF
GRID            10      0.0      0.0     0.0     0       123456
GRID            20      25.      0.0     0.0     0       124
GRID            30      50.      0.0     0.0     0       124
GRID            40      75.      0.0     0.0     0       124
GRID            50      100.     0.0     0.0     0       123456
GRID            60      100.     0.0     10.     0       123456
```

ORIGINAL PAGE IS
OF POOR QUALITY

-91-

3.4.3 Sample Problem Output

Output listing for the three sample problems are shown in sections 3.4.3.1, 3.4.3.2, and 3.4.3.3. Output tape format of the NASTRAN INPUT Loads File is shown in Section 3.3.2.2.

3.4.3.1 Problem #1 Output Listing

Output from sample problem #1 is listed in this section. Only Rigid Body Inertial loads were transformed in this problem.

```
        TRANSFORM PROGRAM DATA
    STPDAT
    CMASS    =      .10000000E+01
    CINERT   =      .10000000E+01
    ISID     =                +0
    XCG      =      .20000000E+01
    YCG      =      .00000000E+00
    ZCG      =      .00000000E+00
    UMF      =  F
    OFFDIG   =  T
    FFREAD   =  T
    TIME     =      .30000000E+02
    CARDS    =  F
    ACCEL    =      .00000000E+00,   .00000000E+00,   .00000000E+00,   .00000000E+00,
                    .00000000E+00,   .00000000E+00
    NUMF     =                +0
    MODINR   =  F
    RIGID    =  T
    APPLYD   =  F
    TRNOUT   =  T
    MASOUT   =  T
    MODLOT   =  T

    $END
    $ACC
    ACCEL    =      .52392689E+03,   .00000000E+00,   .21215325E+02,   .00000000E+00,
                   -.33494918E-03,   .00000000E+00

    $END

     MODAL ACCELERATIONS       .0000      .0000      .0000      .0000      .0000      .0000

      NNODES =    6
      NDF =   36

             IDXYZS =
       11    12    13    14    15    16    21    22    23    24    25    26    31    32    33    34    35    36    41    42
       43    44    45    46    51    52    53    54    55    56    61    62    63    64    65    66

             NODEIDS =
        1     2     3     4     5     6

             XMAT =
       .00      .00      .00      25.00      .00      .00      50.00      .00      .00      75.00      .00      .00
    100.00      .00      .00     100.00      .00    10.00

             DMASS =
      .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00
      .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

             OFFDIAGONAL MASS MATRIX, ROW NO.     1
      .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00
      .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

             OFFDIAGONAL MASS MATRIX, ROW NO.     2
      .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00
```

RUN #1 RIGID BODY LOADS USING SIMULATED USER WEIGHTS FILE

```
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.    3
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.    4
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.    5
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.    6
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.    7
 .00   .00   .00   .00   .00   .00  2.00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.    8
 .00   .00   .00   .00   .00   .00  1.00  2.00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.    9
 .00   .00   .00   .00   .00   .00  1.00  1.00  2.00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.   10
 .00   .00   .00   .00   .00   .00  1.00  1.00  1.00  2.00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.   11
 .00   .00   .00   .00   .00   .00  1.00  1.00  1.00  1.00  2.00   .00   .00   .00   .00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.   12
 .00   .00   .00   .00   .00   .00  1.00  1.00  1.00  1.00  1.00  2.00   .00   .00   .00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.   13
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00  2.00   .00   .00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.   14
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00  1.00  2.00   .00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.   15
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00  1.00  1.00  2.00   .00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00

              OFFDIAGONAL MASS MATRIX, ROW NO.   16
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00  1.00  1.00  1.00  2.00   .00   .00   .00   .00
 .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00   .00
```

ORIGINAL PAGE IS OF POOR QUALITY

40641

94

RUN #1 RIGID BODY LOADS USING SIMULATED USER WEIGHTS FILE (cont'd)

```
OFFDIAGONAL MASS MATRIX, ROW NO.   17
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00 1.00 1.00 1.00 1.00 2.00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   18
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00 1.00 1.00 1.00 1.00 1.00 2.00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   19
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00 2.00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   20
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00 1.00 2.00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   21
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00 1.00 1.00
2.00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   22
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00 1.00 1.00
1.00 2.00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   23
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00 1.00 1.00
1.00 1.00 2.00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   24
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00 1.00 1.00
1.00 1.00 1.00 2.00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   25
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   26
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   27
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   28
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   29
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   30
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

OFFDIAGONAL MASS MATRIX, ROW NO.   31
```

RUN #1 RIGID BODY LOADS USING SIMULATED USER WEIGHTS FILE (cont'd)

```
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

          OFFDIAGONAL MASS MATRIX, ROW NO.   32
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

          OFFDIAGONAL MASS MATRIX, ROW NO.   33
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

          OFFDIAGONAL MASS MATRIX, ROW NO.   34
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

          OFFDIAGONAL MASS MATRIX, ROW NO.   35
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

          OFFDIAGONAL MASS MATRIX, ROW NO.   36
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00
 .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00  .00

    RIGID BODY INERTIA LOADS TRANSFORM  FRB =
     .00      .00      .00      .00      .00      .00-1047.85  -523.93  -566.37  -545.15  -545.15  -545.15-1047.85  -523.93  -566.39
 -545.16  -545.16  -545.16-1047.85  -523.93  -566.41  -545.17  -545.17  -545.17      .00      .00      .00      .00      .00      .00
     .00      .00      .00      .00      .00      .00

  TRB    MATRIX, ROW NUMBER   1
      .25        .00        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00

  TRB    MATRIX, ROW NUMBER   2
      .00        .25        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00

  TRB    MATRIX, ROW NUMBER   3
      .00        .00        .25        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00

  TRB    MATRIX, ROW NUMBER   4
      .25        .00        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00

  TRB    MATRIX, ROW NUMBER   5
      .00        .25        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00        .00        .00        .00        .00
      .00        .00        .00        .00        .00        .00
```

ORIGINAL PAGE IS OF POOR QUALITY

40639

-96-

RUN #1 RIGID BODY LOADS USING SIMULATED USER WEIGHTS FILE

TRB MATRIX, ROW NUMBER 6

.00	.00	.25	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00				

TRB MATRIX, ROW NUMBER 7

.25	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00				

TRB MATRIX, ROW NUMBER 8

.00	.25	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00				

TRB MATRIX, ROW NUMBER 9

.00	.00	.25	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00				

TRB MATRIX, ROW NUMBER 10

.25	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00				

TRB MATRIX, ROW NUMBER 11

.00	.25	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00				

TRB MATRIX, ROW NUMBER 12

.00	.00	.25	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00				

TRB MATRIX, ROW NUMBER 13

.00	.00	.00	.33	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00				

TRB MATRIX, ROW NUMBER 14

.00	.00	.00	.00	.33	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00				

TRB MATRIX, ROW NUMBER 15

.00	.00	.00	.00	.00	.33	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

RUN #1 RIGID BODY LOADS USING SIMULATED USER WEIGHTS FILE (cont'd)

```
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00

TRB      MATRIX, ROW NUMBER  16
         .00       .00       .00       .33       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00

TRB      MATRIX, ROW NUMBER  17
         .00       .00       .00       .00       .33       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00

TRB      MATRIX, ROW NUMBER  18
         .00       .00       .00       .00       .00       .33       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00

TRB      MATRIX, ROW NUMBER  19
         .00       .00       .00       .33       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00

TRB      MATRIX, ROW NUMBER  22
         .00       .00       .00       .00       .00       .00       .50       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00

FORCE          0      108        0     1.00 -523.93      .00       .00

TRB      MATRIX, ROW NUMBER  23
         .00       .00       .00       .00       .00       .00       .00       .50       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00

FORCE          0      108        0     1.00      .00 -261.96      .00

TRB      MATRIX, ROW NUMBER  24
         .00       .00       .00       .00       .00       .00       .00       .00       .50       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00

FORCE          0      108        0     1.00      .00      .00 -283.19

TRB      MATRIX, ROW NUMBER  25
         .00       .00       .00       .00       .00       .00       .50       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00       .00       .00       .00       .00
         .00       .00       .00       .00       .00       .00
```


40637

RUN #1 RIGID BODY LOADS USING SIMULATED USER WEIGHTS FILE (cont'd)

```
FORCE           0     109       0     1.00 -523.93     .00     .00

TRB    MATRIX, ROW NUMBER  26
      .00     .00     .00     .00     .00     .00     .00     .50     .00     .00
      .00     .00     .00     .00     .00     .00     .00     .00     .00     .00
      .00     .00     .00     .00     .00     .00     .00     .00     .00     .00
      .00     .00     .00     .00     .00     .00

FORCE           0     109       0     1.00     .00 -261.96     .00

TRB    MATRIX, ROW NUMBER  27
      .00     .00     .00     .00     .00     .00     .00     .00     .50     .00
      .00     .00     .00     .00     .00     .00     .00     .00     .00     .00
      .00     .00     .00     .00     .00     .00     .00     .00     .00     .00
      .00     .00     .00     .00     .00     .00

FORCE           0     109       0     1.00     .00     .00 -283.19
```

RUN #1 RIGID BODY LOADS USING SIMULATED USER WEIGHTS FILE (cont'd)

3.4.3.2 Problem #2 Output Listing

Output from sample problem #2 is listed in this section. Rigid body inertial loads and modal inertia loads are transformed by this problem.

```
     TRANSFORM PROGRAM DATA
 $TPDAT
 CMASS    =      .10000000E+01
 CINERT   =      .10000000E+01
 ISID     =                 +0
 XCG      =      .20000000E+01
 YCG      =      .00000000E+00
 ZCG      =      .00000000E+00
 UMF      =  T
 OFFDIG   =  T
 FFREAD   =  T
 TIME     =      .30000000E+02
 CARDS    =  F
 ACCEL    =      .00000000E+00,    .00000000E+00,    .00000000E+00,    .00000000E+00,
                 .00000000E+00,    .00000000E+00
 NUMF     =                 +1
 MODINR   =  T
 RIGID    =  T
 APPLYD   =  F
 TRNOUT   =  T
 MASOUT   =  T
 MODLOT   =  T

 $END
 $ACC
 ACCEL    =      .52392689E+03,    .00000000E+00,    .21215325E+02,    .00000000E+00,
                -.33494418E-03,    .00000000E+00

 $END

  MODAL ACCELERATIONS      .0000     .0000     .0000     .0000     .0000     .0000

 NASTOF TAPE GENERATION,FIXED 0DIAGONAL TERMS IN MAA               @@@@@@@@@@@@@@@@@@@@@@@@A

  MODE NO.    1   .FREQ. =     3.6598 -PHIS =
       .61       .01       .00       .32       .01       .00       .10       .01       .00

  MODE NO.    2   .FREQ. =    23.9055 -PHIS =
       .30       .04       .00      -.48       .02      -.00      -.40      -.02      -.00

  MODE NO.    3   .FREQ. =    64.2622 -PHIS =
       .10       .04       .00      -.39      -.02      -.00       .57      -.00       .00

  MODE NO.    4   .FREQ. =   654.3445 -PHIS =
       .39      -.70      -.03      -.22       .36       .02       .10      -.15      -.01

  MODE NO.    5   .FREQ. =   938.0679 -PHIS =
       .19      -.34      -.01       .19      -.40      -.02      -.33       .61       .03

  MODE NO.    6   .FREQ. =  1167.6159 -PHIS =
       .11      -.20      -.01       .32      -.59      -.03       .25      -.50      -.02

  MODE NO.    7   .FREQ. =  1924.4387 -PHIS =
       .24       .28      -.77      -.11      -.14       .37       .04       .06      -.15

  MODE NO.    8   .FREQ. =  2843.6137 -PHIS =
```

RUN #2 RIGID BODY AND MODAL LOADS TRANSFORM USING THE USER MODAL FILE

```
      .11        .13       -.34       .14       .17       -.45       -.20       -.25       .65
 MODE NO.    9  ,FREQ. = 3553.4060 -PHIS =
      .06        .08       -.20       .20       .24       -.64       .17        .20       -.55
         IDXYZS =
   23    25    26    33    35    36    43    45    46
         NODEIDS =
    1     2     3     4     5     6
         XMAT =
    .00       .00       .00       25.00      .00      .00      50.00      .00      .00      75.00     .00     .00
 100.00       .00       .00      100.00      .00    10.00
         DMASS =
 2.00  2.00  2.00  2.00  2.00  2.00  2.00  2.00  2.00
         OFFDIAGONAL MASS MATRIX, ROW NO.    1
 2.00  1.00  1.00   .00   .00   .00   .00   .00   .00
         OFFDIAGONAL MASS MATRIX, ROW NO.    2
 1.00  2.00  1.00   .00   .00   .00   .00   .00   .00
         OFFDIAGONAL MASS MATRIX, ROW NO.    3
 1.00  1.00  2.00   .00   .00   .00   .00   .00   .00
         OFFDIAGONAL MASS MATRIX, ROW NO.    4
  .00   .00   .00  2.00  1.00  1.00   .00   .00   .00
         OFFDIAGONAL MASS MATRIX, ROW NO.    5
  .00   .00   .00  1.00  2.00  1.00   .00   .00   .00
         OFFDIAGONAL MASS MATRIX, ROW NO.    6
  .00   .00   .00  1.00  1.00  2.00   .00   .00   .00
         OFFDIAGONAL MASS MATRIX, ROW NO.    7
  .00   .00   .00   .00   .00   .00  2.00  1.00  1.00
         OFFDIAGONAL MASS MATRIX, ROW NO.    8
  .00   .00   .00   .00   .00   .00  1.00  2.00  1.00
         OFFDIAGONAL MASS MATRIX, ROW NO.    9
  .00   .00   .00   .00   .00   .00  1.00  1.00  2.00
   RIGID BODY INERTIA LOADS TRANSFORM  FRB =
 -42.45  -21.22  -21.22  -42.46  -21.23  -21.23  -42.48  -21.24  -21.24
TRB     MATRIX, ROW NUMBER    1
     .25       .00       .00       .00       .00       .00       .00       .00       .00
FORCE          0     101      0     1.00  -10.61     .00      .00
TRB     MATRIX, ROW NUMBER    2
     .00       .25       .00       .00       .00       .00       .00       .00       .00
FORCE          0     101      0     1.00     .00   -5.31      .00
```


108

RUN #2 RIGID BODY AND MODAL LOADS TRANSFORM USING THE USER MODAL FILE (cont'd)

```
TRB    MATRIX, ROW NUMBER    3
     .00        .00        .25        .00        .00        .00        .00        .00        .00

FORCE          0     101        0     1.00       .00       .00     -5.31

TRB    MATRIX, ROW NUMBER    4
     .25        .00        .00        .00        .00        .00        .00        .00        .00

FORCE          0     102        0     1.00    -10.61       .00       .00

TRB    MATRIX, ROW NUMBER    5
     .00        .25        .00        .00        .00        .00        .00        .00        .00

FORCE          0     102        0     1.00       .00     -5.31       .00

TRB    MATRIX, ROW NUMBER    6
     .00        .00        .25        .00        .00        .00        .00        .00        .00

FORCE          0     102        0     1.00       .00       .00     -5.31

TRB    MATRIX, ROW NUMBER    7
     .25        .00        .00        .00        .00        .00        .00        .00        .00

FORCE          0     103        0     1.00    -10.61       .00       .00

TRB    MATRIX, ROW NUMBER    8
     .00        .25        .00        .00        .00        .00        .00        .00        .00

FORCE          0     103        0     1.00       .00     -5.31       .00

TRB    MATRIX, ROW NUMBER    9
     .00        .00        .25        .00        .00        .00        .00        .00        .00

FORCE          0     103        0     1.00       .00       .00     -5.31

TRB    MATRIX, ROW NUMBER   10
     .25        .00        .00        .00        .00        .00        .00        .00        .00

FORCE          0     104        0     1.00    -10.61       .00       .00

TRB    MATRIX, ROW NUMBER   11
     .00        .25        .00        .00        .00        .00        .00        .00        .00

FORCE          0     104        0     1.00       .00     -5.31       .00

TRB    MATRIX, ROW NUMBER   12
     .00        .00        .25        .00        .00        .00        .00        .00        .00

FORCE          0     104        0     1.00       .00       .00     -5.31

TRB    MATRIX, ROW NUMBER   13
     .00        .00        .00        .33        .00        .00        .00        .00        .00

FORCE          0     105        0     1.00    -14.14       .00       .00

TRB    MATRIX, ROW NUMBER   14
     .00        .00        .00        .00        .33        .00        .00        .00        .00
```

RUN #2 RIGID BODY AND MODAL LOADS TRANSFORM USING THE USER MODAL FILE (cont'd)

```
FORCE          0      105        0     1.00      .00   -7.07      .00

TRB     MATRIX, ROW NUMBER  15
     .00       .00       .00       .00       .00       .33       .00       .00       .00

FORCE          0      105        0     1.00      .00     .00    -7.07

TRB     MATRIX, ROW NUMBER  16
     .00       .00       .00       .33       .00       .00       .00       .00       .00

FORCE          0      106        0     1.00   -14.14     .00      .00

TRB     MATRIX, ROW NUMBER  17
     .00       .00       .00       .00       .33       .00       .00       .00       .00

FORCE          0      106        0     1.00      .00   -7.07      .00

TRB     MATRIX, ROW NUMBER  18
     .00       .00       .00       .00       .00       .33       .00       .00       .00

FORCE          0      106        0     1.00      .00     .00    -7.07

TRB     MATRIX, ROW NUMBER  19
     .00       .00       .00       .33       .00       .00       .00       .00       .00

FORCE          0      107        0     1.00   -14.14     .00      .00

TRB     MATRIX, ROW NUMBER  22
     .00       .00       .00       .00       .00       .00       .50       .00       .00

FORCE          0      108        0     1.00   -21.24     .00      .00

TRB     MATRIX, ROW NUMBER  23
     .00       .00       .00       .00       .00       .00       .00       .50       .00

FORCE          0      108        0     1.00      .00  -10.62      .00

TRB     MATRIX, ROW NUMBER  24
     .00       .00       .00       .00       .00       .00       .00       .00       .50

FORCE          0      108        0     1.00      .00     .00   -10.62

TRB     MATRIX, ROW NUMBER  25
     .00       .00       .00       .00       .00       .00       .50       .00       .00

FORCE          0      109        0     1.00   -21.24     .00      .00

TRB     MATRIX, ROW NUMBER  26
     .00       .00       .00       .00       .00       .00       .00       .50       .00

FORCE          0      109        0     1.00      .00  -10.62      .00

TRB     MATRIX, ROW NUMBER  27
     .00       .00       .00       .00       .00       .00       .00       .00       .50

FORCE          0      109        0     1.00      .00     .00   -10.62
```


104

RUN #2 RIGID BODY AND MODAL LOADS TRANSFORM USING THE USER MODAL FILE

```
MODAL INERTIA LOADS TRANSFORM   FMOD =
 .00     .00     .00     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER   1
     .25     .00     .00     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER   2
     .00     .25     .00     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER   3
     .00     .00     .25     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER   4
     .25     .00     .00     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER   5
     .00     .25     .00     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER   6
     .00     .00     .25     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER   7
     .25     .00     .00     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER   8
     .00     .25     .00     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER   9
     .00     .00     .25     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER  10
     .25     .00     .00     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER  11
     .00     .25     .00     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER  12
     .00     .00     .25     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER  12
     .00     .00     .25     .00     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER  13
     .00     .00     .00     .33     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER  14
     .00     .00     .00     .00     .33     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER  15
     .00     .00     .00     .00     .00     .33     .00     .00     .00

TMOD   MATRIX, ROW NUMBER  16
     .00     .00     .00     .33     .00     .00     .00     .00     .00

TMOD   MATRIX, ROW NUMBER  17
     .00     .00     .00     .00     .33     .00     .00     .00     .00
```

RUN #2 RIGID BODY AND MODAL LOADS TRANSFORM USING THE USER MODAL FILE (cont'd)

```
THOD    MATRIX, ROW NUMBER  18
      .00        .00        .00        .00        .00        .33        .00        .00        .00

THOD    MATRIX, ROW NUMBER  19
      .00        .00        .00        .33        .00        .00        .00        .00        .00

THOD    MATRIX, ROW NUMBER  17
      .00        .00        .00        .00        .33        .00        .00        .00        .00

THOD    MATRIX, ROW NUMBER  18
      .00        .00        .00        .00        .00        .33        .00        .00        .00

THOD    MATRIX, ROW NUMBER  22
      .00        .00        .00        .00        .00        .00        .50        .00        .00

THOD    MATRIX, ROW NUMBER  23
      .00        .00        .00        .00        .00        .00        .00        .50        .00

THOD    MATRIX, ROW NUMBER  24
      .00        .00        .00        .00        .00        .00        .00        .00        .50

THOD    MATRIX, ROW NUMBER  25
      .00        .00        .00        .00        .00        .00        .50        .00        .00

THOD    MATRIX, ROW NUMBER  26
      .00        .00        .00        .00        .00        .00        .00        .50        .00
DATA IGNORED - IN CONTROL MODE
```

RUN #2 RIGID BODY AND MODAL LOADS TRANSFORM USING THE USER MODAL FILE (cont'd)

```
@EOF
@EOF IGNORED - IN CONTROL MODE

@PMD

PMD 0029-12/24- 12:31:57
SYS$*RLIBS. LEVEL 69

BANK- $IBANK        SEGMENT- A                ELEMENT- MAIN              $( 01)  AT ADDRESS  015451   CREATED ON: 24 DEC 74 AT 12:28:45
000000  015453   745660007477  000000000000  745660007502  000100072301  000000072200  000131000000  745660013754  000100072273
000010  015463   000134000000  720400013533  745660011326  000100072302  000000072200  000137000000  745660014143  000000072173
000020  015473   000000072303  000000072304  000000072137  000000072177  000000072305  000142000000  100000072175  744000015505
000030  015503   745660040037  000144000000  100000072174  744000015511  745660040026  000146000000  100000072176  744000015515
000040  015513   745660040020  000150000000  745660014143  000000072173  000000072306  000000072306  000000072305  000151000000
000050  015523   745660011217  000000072307  745660011217  000000072310

BANK- $IBANK        SEGMENT- A                ELEMENT- TRANS             $( 01)  AT ADDRESS  015527   CREATED ON: 24 DEC 74 AT 12:29:05
000000  015527   107400777776  140013200001  010000073420  745660014143  000000072756  000000073377  000142073415  107400777762
000010  015537   010000072761  107440777776  140040045611  270020073400  010040000117  237740000002  050001446204  702340015545
000020  015547   107400001127  722400000001  020000000001  702360015544  745660007502  000100073401  000000072776  000155073415
000030  015557   100000072754  150000073421  510000000014  742000015571  745660013756  000100073045  000162073415  720400013533
000040  015567   745660011217  000000073402  107400777776  140000045611  107740000001  010000000117  010340072762  270020073403
000050  015577   230340073420  050001473066  702340015600  745660007502  000200073401  000000073017  000176073415  000002016003
000060  015607   270020072760  060020073422  100000044645  744000015626  745660013756  000100073056  000202073415  000020072754
000070  015617   000020072760  270020073403  230340073420  107401473066  720440013530  702340015622  720400013533  050000072764
```

3.4.3.3 Problem #3 Output Listing

The OUTPUT listing is for the applied force transform, which was conducted by deleting the calls to RBITRN and MILTRN and inputting the applied force array by means of a DATA statement.

```
   TRANSFORM PROGRAM DATA
STPDAT
CMASS    =      .10000000E+01
CINERT   =      .10000000E+01
ISID     =                  +0
XCG      =      .00000000E+00
YCG      =      .00000000E+00
ZCG      =      .00000000E+00
UMF      =  F
OFFDIG   =  F
FFREAD   =  F
TIME     =      .00000000E+00
CARDS    =  F
ACCEL    =      .00000000E+00,   .00000000E+00,   .00000000E+00,   .00000000E+00,
                .00000000E+00,   .00000000E+00
NUMF     =                  +0
MODINR   =  F
RIGID    =  F
APPLYD   =  T
TRNOUT   =  T
MASOUT   =  T
MODLOT   =  T

SEND

   APPLIED FORCE TRANSFORM    FAP =
  .00     .00  200.00     .00 -100.00 1000.00  100.00      .00      .00

TAP    MATRIX, ROW NUMBER   1
      .25     .00     .00     .00     .00     .00     .00     .00     .00

TAP    MATRIX, ROW NUMBER   2
      .00     .25     .00     .00     .00     .00     .00     .00     .00

TAP    MATRIX, ROW NUMBER   3
      .00     .00     .25     .00     .00     .00     .00     .00     .00

FORCE          0     101       0    1.00     .00     .00   50.00

TAP    MATRIX, ROW NUMBER   4
      .25     .00     .00     .00     .00     .00     .00     .00     .00

TAP    MATRIX, ROW NUMBER   5
      .00     .25     .00     .00     .00     .00     .00     .00     .00

TAP    MATRIX, ROW NUMBER   6
      .00     .00     .25     .00     .00     .00     .00     .00     .00

FORCE          0     102       0    1.00     .00     .00   50.00

TAP    MATRIX, ROW NUMBER   7
      .25     .00     .00     .00     .00     .00     .00     .00     .00

TAP    MATRIX, ROW NUMBER   8
      .00     .25     .00     .00     .00     .00     .00     .00     .00
```

RUN #3 APPLIED FORCE TRANSFORM USING CARD INPUT

```
TAP     MATRIX, ROW NUMBER    9
      .00       .00       .25       .00       .00       .00       .00       .00       .00

FORCE           0      103      0    1.00      .00      .00   50.00

TAP     MATRIX, ROW NUMBER   10
      .25       .00       .00       .00       .00       .00       .00       .00       .00

TAP     MATRIX, ROW NUMBER   11
      .00       .25       .00       .00       .00       .00       .00       .00       .00

TAP     MATRIX, ROW NUMBER   12
      .00       .00       .25       .00       .00       .00       .00       .00       .00

FORCE           0      104      0    1.00      .00      .00   50.00

TAP     MATRIX, ROW NUMBER   13
      .00       .00       .00       .25       .00       .00       .00       .00       .00

TAP     MATRIX, ROW NUMBER   14
      .00       .00       .00       .00       .33       .00       .00       .00       .00

FORCE           0      105      0    1.00      .00   -33.30      .00

TAP     MATRIX, ROW NUMBER   15
      .00       .00       .00       .00       .00       .25       .00       .00       .00

FORCE           0      105      0    1.00      .00      .00  250.00

TAP     MATRIX, ROW NUMBER   16
      .00       .00       .00       .50       .00       .00       .00       .00       .00

TAP     MATRIX, ROW NUMBER   17
      .00       .00       .00       .00       .33       .00       .00       .00       .00

FORCE           0      106      0    1.00      .00   -33.30      .00

TAP     MATRIX, ROW NUMBER   18
      .00       .00       .00       .00       .00       .50       .00       .00       .00

FORCE           0      106      0    1.00      .00      .00  500.00

TAP     MATRIX, ROW NUMBER   19
      .00       .00       .00       .25       .00       .00       .00       .00       .00

TAP     MATRIX, ROW NUMBER   20
      .00       .00       .00       .00       .33       .00       .00       .00       .00

FORCE           0      107      0    1.00      .00   -33.30      .00

TAP     MATRIX, ROW NUMBER   21
      .00       .00       .00       .00       .00       .25       .00       .00       .00

FORCE           0      107      0    1.00      .00      .00  250.00

TAP     MATRIX, ROW NUMBER   22
      .00       .00       .00       .00       .00       .00       .50       .00       .00
```

RUN #3 APPLIED FORCE TRANSFORM USING CARD INPUT (cont'd)

```
FORCE          0     108       0    1.00   50.00     .00     .00

TAP    MATRIX, ROW NUMBER  23
     .00      .00      .00      .00      .00      .00      .00      .50      .00

TAP    MATRIX, ROW NUMBER  24
     .00      .00      .00      .00      .00      .00      .00      .00      .50

TAP    MATRIX, ROW NUMBER  25
     .00      .00      .00      .00      .00      .00      .50      .00      .00

FORCE          0     109       0    1.00   50.00     .00     .00

TAP    MATRIX, ROW NUMBER  26
     .00      .00      .00      .00      .00      .00      .00      .50      .00

TAP    MATRIX, ROW NUMBER  27
     .00      .00      .00      .00      .00      .00      .00      .00      .50

**NTRAN ERROR** UNIT 10 HAS IMPROPER DEVICE.
  I/O CALLED AT SEQUENCE NUMBER 000151 OF MAIN PROGRAM

**NTRAN ERROR** UNIT 10 HAS IMPROPER DEVICE.
  I/O CALLED AT SEQUENCE NUMBER 000151 OF MAIN PROGRAM


@EOF
@EOF IGNORED - IN CONTROL MODE

@PMD,E

PMD 0029-12/11- 14:17:03
@FIN
```

RUN #3 APPLIED FORCE TRANSFORM USING CARD INPUT (cont'd)

4.0 SUBROUTINE DOCUMENTATION

The TRAIL Program was written in Fortran V to be used on a Univac 1100 series digital computer with an Exec-VIII system. It has been designed to be used within the Integrated Structural Analysis System (ISAS) for the purpose of transforming inertial and applied loads on a dynamic loads model to a finite element model having many more node points. In the programs present form the maximum number of degrees-of-freedom in the finite element model is 410. The storage core capatity the program occupies with this maximum capability is approximately 57K words.

4.1 Main Program Description

The controlling routine within TRAIL is MAIN. This routine manages the program calculations and the program output. The structure of the calling sequence is simple and can be visualized by observing the routine listing. Figure 2-1 of this document illustrates the program flow. Main reads the transform logic data, writes the header information on the NASTRAN Input Loads File (NASFIL), calls the requested transformation routine, and outputs the requested printed and magnetic tape data.

A Fortran Procedure named DIM was used to easily change maximum array sizes. It is intended to allow for decreasing the required core size when running problems which do not require the maximum default dimensions. This Procedure is described in Section 4.1.3.

-111-

4.1.1 Routine MAIN Characteristics

OUTLINE FOR SCIENTIFIC SUBPROGRAM DOCUMENTATION

IDENTIFICATION

Name/Title	MAIN
Programmer/Date	F. STRATMAN 4/74
Author/Date	F. STRATMAN 5/74
Organization/Installation	LEC-ASD FOR SMD
Machine Identification	UNIVAC 1110
Source Language	FORTRAN V

STORAGE:

IBANK 54_8

DBANK 152_8

INPUT VARIABLES:

SYMBOL	TYPE	SIZE	DESCRIPTION
A	A	1	Title of run
TRNOUT	L	1	Printout control for transformation matrices
MASOUT	L	1	Printout control for mass matrix
MODLOT	L	1	Printout control for user model data
MODINR	L	1	Computation control for modal inertia loads
RIGID	L	1	Computation control for rigid body loads transform
CARDS	L	1	Card punch output control
APPLYD	L	1	Computation control for applied loads transform
NASFIL	I	1	Unit number of NASTRAN input loads file
UMF	L	1	Indicator for User Modal File
OFFDIG	L	1	Indicator for off-diagonal mass matrix
FFREAD	L	1	Indicator for Flight Conditions File read
CMASS	R	1	Scale factor for masses
CINERT	R	1	Scale factor for inertias
ISID	I	1	Set identification number for NASTRAN force and moment cards
XCG	R	1	
YCG	R	1	Cartesian coordinate location of vehicle c.g.
ZCG	R	1	
TIME	R	1	Flight time for loads and accelerations
ACCEL	R	6	Rigid body accelerations
NUMF	I	1	File location on modal input tape

CARD OR TERMINAL INPUT:

The Namelist $TPDAT was used to input program control parameters to the main routine. $TPDAT was input with cards for the checkout case. Other means of inputting this type of data could be used such as the Exec-8 7/8ADD command. A description of Namelist $TPDAT follows:

NAMELIST/TPDAT/CMASS,CINERT,ISID,XCG,YCG,ZCG,UMF,OFFDIG,FFREAD,
TIME,CARDS,ACCEL,NUMF,MODINR,RIGID,APPLYD,TRNOUT,MASOUT,MODLOT

LIBRARY ROUTINES REQUIRED:

The NTRAN routine is required by routine MAIN to write the NASTRAN Input File (NASFIL). Appendix A has a description of this routine.

LISTING:

A listing of routine MAIN can be found in Section 4.1.2.

4.1.2 LISTING OF ROUTINE MAIN

```
@FOR,S   MAIN,MAIN
FOR SE1X-12/23/74-21:49:54 (0,)

   MAIN PROGRAM

   STORAGE USED: CODE(1) 000541; DATA(0) 001521; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   TPINFO 000016
    0004   FACTOR 000004
    0005   DATOUT 000003

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006   NTRAN
    0007   RBITRN
    0010   MILTRN
    0011   AFTRAN
    0012   NINTR$
    0013   NRNL$
    0014   NPRT$
    0015   NIO2$
    0016   NWNL$
    0017   NSTOP$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   000134 1F       0000 R 000000 A        0003 R 000007 ACCEL    0000 L 000037 APPLYD   0004 L 000003 CARDS
    0004 R 000001 CINERT   0004 R 000000 CMASS    0003 L 000005 FFREAD   0004 I 000002 ISID     0000 I 000040 L
    0005 L 000001 MASOUT   0000 L 000035 MODINR   0005 L 000002 MODLOT   0000 I 000034 NASFIL   0003 I 000015 NUMF
    0003 L 000004 OFFDIG   0000 L 000036 RIGID    0003 R 000006 TIME     0000   000041 TPDAT    0005 L 000000 TRNOUT
    0003 L 000003 UMF      0003 R 000000 XCG      0003 R 000001 YCG      0003 R 000002 ZCG


00101     1*          COMPILER (DATA = SHORT)                                              000000
00103     2*          DIMENSION A(28)                                                      000001
00105     3*          DATA NASFIL /10/                                                     000001
00107     4*          DATA A /'NASTRAN INPUT LOADS FILE' /                                 000001
00111     5*          LOGICAL TRNOUT/.TRUE. /,MASOUT/.TRUE. /,MODLOT/.TRUE. /              000001
00115     6*          LOGICAL MODINR/.TRUE. /,RIGID /.TRUE. /,APPLYD/.TRUE. /,CARDS        000001
00115     7*         *    /.TRUE./   ,UMF,OFFDIG,FFREAD                                    000001
00122     8*          DATA CMASS,CINERT/ 2 * 1./                                           000001
00125     9*          COMMON/TPINFO/XCG,YCG,ZCG,UMF,OFFDIG,FFREAD,TIME,ACCEL(6),NUMF       000001
00126    10*          COMMON/FACTOR/CMASS,CINERT,ISID,CARDS                                000001
00127    11*          COMMON /DATOUT/  TRNOUT,MASOUT,MODLOT                                000001
00130    12*          NAMELIST/TPDAT/CMASS,CINERT,ISID,XCG,YCG,ZCG,UMF,OFFDIG,FFREAD,      000001
00130    13*         *           TIME,CARDS,ACCEL   ,NUMF,MODINR,RIGID,APPLYD              000001
00130    14*         *,               TRNOUT,MASOUT,MODLOT                                 000001
00130    15*      C                                                                        000001
00131    16*          READ (5,TPDAT)                                                       000001
00134    17*          PRINT 1                                                              000005
00136    18*        1 FORMAT (1H1/ 5X, 'TRANSFORM PROGRAM DATA')                           000011
```

4.1.2 LISTING OF ROUTINE MAIN (cont'd)

```
00137      WRITE (6,TPDAT)                         000011
00142      CALL NTRAN (NASFIL,1,28,A,L,22)         000015
00143      IF (RIGID )  CALL RBITRN                000025
00145      IF (MODINR)  CALL MILTRN                000031
00147      IF (APPLYD)  CALL AFTRAN                000035
00151      CALL NTRAN(NASFIL,9,9,22)               000041
00152      STOP                                    000047
00153      END                                     000053

     END OF COMPILATION:        NO  DIAGNOSTICS.
```

4.1.3 Description of Fortran Proc, Dim

OUTLINE FOR SCIENTIFIC SUBPROGRAM DOCUMENTATION

IDENTIFICATION

Name/Title	FORTRAN PROCEDURE DIM
Programmer/Date	FRED STRATMAN 4/74
Author/Date	FRED STRATMAN 10/74
Organization/Installation	LEC-ASD FOR SMD
Machine Identification	UNIVAC 1110
Source Language	FORTRAN V

PURPOSE

DIM sets dimensions for the larger arrays used in the program.

USAGE

- Calling Sequence

INCLUDE DIM, LIST

PARAMETER DEFINITION

Parameter Name	Dimension	Type	Description
MAXDF	410	I	Maximum degrees-of-freedom
MAXMOD	100	I	Maximum elastic modes
MAXNOD	120	I	Maximum nodes

-116-

```
@PDP,LF DIM,DIM

PDP-8A-07/17-03:02

PE0001        DIM   PROC
  0002        C*     ARRAY DIMENSIONS
  0003        C
  0004               PARAMETER MAXDI  = 410
  0005               PARAMETER MAXNOD = 100
  0006               PARAMETER MAXNOL = 120
  0007        C
  0008         END

END PDP
```


-117-

4.2 Common Block Information

There are four named common blocks used within TRAIL. A common block versus subroutine matrix for the common blocks used is shown in Figure 4-1.

4.2.1 List of Common Block Variables

A list of the contents of each common block follows.

/TPINFO/XCG,YCG,ZCG,UMF,OFFDIG,FFREAD,TIME,ACCEL,NUMF
/FACTOR/CMASS,CINERT,ISID,CARDS
/DATOUT/TRNOUT,MASOUT,MODLOT
/MODAL/A,NDF,NMODES,IDXYZ,NNODES,NODEID,XMAT,NDFFE,
FREQ,QDD,DMASS,RMASS,PHI,IUMF,MASOFF

	TPINFO	FACTOR	DATOUT	MODAL
AFTRAN	X		X	
MAIN	X	X	X	
MILTRN				X
RBITRN	X		X	X
RDUMF	X		X	X
TRANS		X	X	X

FIGURE 4-1

Common Block Cross Reference Table

-118-

4.3 Description of Subroutine AFTRAN

OUTLINE FOR SCIENTIFIC SUBPROGRAM DOCUMENTATION

IDENTIFICATION

Name/Title	SUBROUTINE AFTRAN
Programmer/Date	FRED STRATMAN 4/74
Author/Date	FRED STRATMAN 5/74
Organization/Installation	LEC-ASD FOR SMD
Machine Identification	UNIVAC 1110
Source Language	FORTRAN V

PURPOSE

AFTRAN sets up an array of applied forces and moments.

USAGE

- Calling Sequence

Call AFTRAN

MODEL

AFTRAN copies the external forces and moments array, [FAP], from the Compressed Aerodynamic Forces and Conditions File described in Section 3.2.3.3.

NTRAN operations are used to skip to the desired data block and to read the data. A flow of AFTRAN operations is shown in Figure 4-2.

LISTING

A listing of the AFTRN subroutine is found in Section 4.3.1.

-119-

• Symbol Definition

Coding	Type	Description	Units
ICAFCF	I	Unit number of Compressed Aero Forces and Conditions File	
NENG	I	Number of thrust forces	
NEXT	I	Number of applied forces	
NTIMES	I	Number of time points on file	
TAPCK	H	Name passed to transform routine to verify that TAP matrix is read	
TSTART } TSTOP	R	Start and stop times of simulation on ICAFCF	
FAP	R	Array of applied forces and moments	

SUBROUTINE AFTRAN

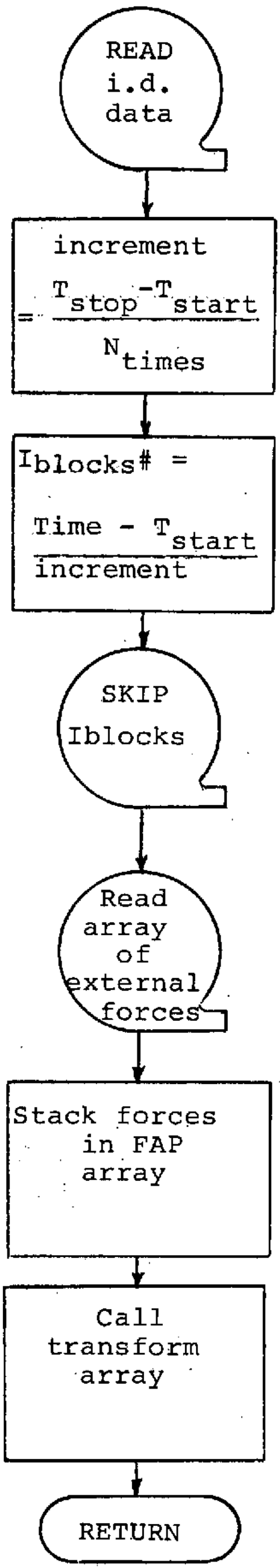


FIGURE 4-2

Flow Chart of Subroutine AFTRAN

-121-

4.3.1 Listing of Subroutine AFTRAN

```
@FOR,S   AFTRAN,AFTRAN
FOR S E1X-12/23/74-21:45:08 (0,)

   SUBROUTINE AFTRAN    ENTRY POINT 000160

   STORAGE USED: CODE(1) 000167; DATA(0) 000617; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   DATOUT 000003
    0004   TPINFO 000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0005   NTRAN
    0006   TRANS
    0007   NPRT$
    0010   NI01$
    0011   NI02$
    0012   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   000471 1F        0001   000123 136G      0001   000133 144G      0004   000007 ACCEL     0000 R 000543 BUFF
    0000 R 000000 FAP       0004   000005 FFREAD    0000 I 000470 I         0000 I 000466 IBLOCK    0000 I 000503 IBUFF
    0000 I 000455 ICAFCF    0000 I 000465 INCRMT    0000   000602 INJP$     0000 I 000456 L         0000 L 000503 LBUFF
    0000 I 000467 M         0003 L 000001 MASOUT    0003 L 000002 MODLOT    0000 I 000464 N         0000 I 000457 NENG
    0000 I 000460 NEXT      0000 I 000463 NTIMES    0004   000016 NUMF      0004   000004 OFFDIG    0000 I 000454 TAPCK
    0004 R 000006 TIME      0003 L 000000 TRNOUT    0000 R 000461 TSTART    0000 R 000462 TSTOP     0004   000003 UNF
    0004   000000 XCG       0004   000001 YCG       0004   000002 ZCG
00000  *DIAGNOSTIC*  THE NAME EXTXYZ APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00000  *DIAGNOSTIC*  THE NAME IDEXT APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*             SUBROUTINE AFTRAN                                                     000000
00101      2*      C*      A ROUTINE TO TRANSFORM APPLIED LOADS                                 000000
00103      3*             DIMENSION BUFF(56),EXTXYZ(3,6),IDEXT(36),FAP(300)                     000000
00104      4*             DIMENSION IBUFF(56)                                                   000000
00105      5*             COMMON /DATOUT/  TRNOUT,MASOUT,MODLOT                                 000000
00106      6*             COMMON/TPINFO/XCG,YCG,ZCG,UNF,OFFDIG,FFREAD,TIME,ACCEL(6),NUMF        000000
00107      7*             LOGICAL TRNOUT,MASOUT,MODLOT                                          000000
00110      8*             LOGICAL LBUFF(56)                                                     000000
00111      9*             EQUIVALENCE (BUFF(1),IBUFF(1),LBUFF(1))                               000000
00112     10*             INTEGER TAPCK/6HTAP   /                                               000000
00114     11*             DATA ICAFCF/3/                                                        000000
00116     12*             CALL NTRAN(ICAFCF,10)                                                 000000
00116     13*      C*     READ ID INFORMATION                                                   000000
00117     14*             CALL NTRAN(ICAFCF,2,37,BUFF,L,22)                                     000003
00120     15*             NENG  =IBUFF(28)                                                      000013
00121     16*             NEXT  =IBUFF(29)                                                      000015
00122     17*             TSTART= BUFF(32)                                                      000017
```

```
00123                 TSTOP = BUFF(33)                                                        000021
00124                 NTIMES =IBUFF(34)                                                       000023
00125                 N = NENG + NEXT                                                         000025
00125          C*     READ EXTERNAL FORCE LOCATIONS                                           000025
00126                 CALL NTRAN (ICAFCF,2,N,BUFF,L,22)                                       000031
00126          C*     FIND BLOCK FOR DESIRED FLIGHT CONDITIONS                                000031
00127                 INCRMT =(TSTOP-TSTART)/NTIMES                                           000043
00130                 IBLOCK = (TIME-TSTART)/INCRMT                                           000056
00131                 CALL NTRAN(ICAFCF,7,IBLOCK)                                             000072
00131          C*     READ ARRAY OF FORCES                                                    000072
00132                 N = N + 15                                                              000077
00133                 CALL NTRAN(ICAFCF,2,N,BUFF,L,22)                                        000102
00134                 M = 15 + N                                                              000112
00135                 DO 30 I=1,NEXT                                                          000115
00140              30 FAP(I) = BUFF(M + I)                                                    000123
00142                 PRINT 1, (FAP(I), I=1,NEXT)                                             000125
00150               1 FORMAT ('0', '    APPLIED FORCE TRANSFORM   FAP =' , 300(/15F8.2))     000136
00151                 CALL TRANS(FAP,NEXT,TAPCK)                                              000136
00152                 RETURN                                                                  000143
00153                 END                                                                     000166

      END OF COMPILATION:          2 DIAGNOSTICS.
```

OUTLINE FOR SCIENTIFIC SUBPROGRAM DOCUMENTATION

IDENTIFICATION

Name/Title	SUBROUTINE MILTRN
Programmer/Date	FRED STRATMAN 4/74
Author/Date	FRED STRATMAN 5/74
Organization/Installation	LEC-ASD FOR SMD
Machine Identification	UNIVAC 1110
Source Language	FORTRAN V

PURPOSE

MILTRN calculates an array of modal inertia loads acting at the degrees of freedom of the loads model.

USAGE

- Calling Sequence

Call MILTRN

- Symbol Definition

Coding	Type	Description	Units
FMOD(NDF)	R	The column matrix of modal inertia loads	
MASS(NDF,NDF)	R	The mass matrix	
NDF	I	Number of degrees-of-freedom	
NODACC(NDF)	R	Acceleration of gridpoint nodes at each d.o.f.	
PHI(NDF, NMODES)	R	Modal matrix	

Coding	Type	Description	Units
QDD(NMODES)	R	Modal accelerations	
TMDCK	H	Name passed to transform routine to insure that TMOD matrix is read	

- Model

To compute the FMOD array the nodal accelerations must first be computed using the PHI array from the User Modal File and QDD from the Flight Conditions File in this manner.

$$\left\{ NODACC \right\} = NDF \overset{\leftarrow NMODES \rightarrow}{\left[PHI \right]} \left\{ QDD \right\} \; NMODES$$

The modal MASS matrix which premultiplies nodal accelerations to obtain FMOD is an NDF X NDF array which is symmetrical, and can be a diagonal matrix.

$$\left\{ FMOD \right\} = \left[-MASS \right] \left\{ NODACC \right\}$$

LISTING

A listing of subroutine MILTRN is found in Section 4.4.1.

-125-

4.4.1 Listing of Subroutine MILTRN

```
@FOR,S MILTRN,MILTRN
FOR S£1X-12/23/74-21:45:02 (0,)

    SUBROUTINE MILTRN      ENTRY POINT 000131

    STORAGE USED: CODE(1) 000142; DATA(0) 001529; BLANK COMMON(2) 000000

     COMMON BLOCKS:

     0003   MODAL  123531

    EXTERNAL REFERENCES (BLOCK, NAME)

     0004   TRANS
     0005   NRES$
     0006   NRBU$
     0007   NIO1$
     0010   NIO2$
     0011   NPRT$
     0012   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001    000016 116G      0001    000021 122G      0001    000042 131G      0001    000053 134G      0001    000043 143G
     0001    000102 152G      0000    001467 3F        0003    000000 A         0003    001773 DMASS     0000 R 000632 FMOD
     0003    001626 FREQ      0000 I 001465 I         0003    000032 IDXYZ     0000    001505 INJP$     0003    123527 IUMF
     0000 I 001466 J         0003 I 123530 MASOFF    0003 R 002625 MASS      0003 I 000030 NDF       0003    001625 NDFFE
     0003 I 000031 NMODES    0003    000669 NNODES    0000 R 000000 NODACC    0003    000465 NODEID    0003 R 003457 PHI
     0003 R 001627 QOD       0003    002625 RMASS     0000 I 001464 TMDCK     0003    001055 XMAT

 00101      1*            SUBROUTINE MILTRN                                                000000
 00101      2*      C                                                                      000000
 00101      3*      C*  ROUTINE TO SET UP TRANSFORMATION OF MODAL INERTIA LOADS            000000
 00101      4*      C                                                                      000000
 00103      5*            INCLUDE  DIM,LIST                                                000000
 00103      5*      DIM  PROC                                                              000000
 00103      5*      C*    ARRAY DIMENSIONS                                                 000000
 00103      5*      C                                                                      000000
 00104      5*            PARAMETER MAXDF = 410                                            000000
 00105      5*            PARAMETER MAXMOD = 100                                           000000
 00106      5*            PARAMETER MAXNOD = 120                                           000000
 00106      5*      C                                                                      000000
 00106      5*       END                                                                   000000
 00107      6*            COMMON /MODAL/A(24),NDF,NMODES,IDXYZ(MAXDF),NNODES,NODEID(MAXNOD),    000000
 00107      7*           *XMAT(MAXNOD,3),NDFFE,FREQ,QDD(MAXMOD),DMASS(MAXDF),RMASS(MAXDF),     000000
 00107      8*           *PHI(MAXDF,MAXMOD),IUMF,MASOFF                                   000000
 00110      9*            DIMENSION NODACC(MAXDF),FMOD(MAXDF)                              000000
 00111     10*            REAL MASS(MAXDF),NODACC                                          000000
 00112     11*            EQUIVALENCE (MASS(1),RMASS(1))                                   000000
 00113     12*            INTEGER TMDCK/6HTMOD  /                                          000000
```

```
00113    C*      CALCULATE NODAL ACCELERATIONS                                  000000
00115          DO 120 I=1,NDF                                                   000000
00120          NODACC(I) = 0.                                                   000016
00121          DO 120 J=1,NMODES                                                000021
00124      120 NODACC(I) = NODACC(I) + PHI(I,J)*QDD(J)                          000021
00124    C                                                                      000021
00124    C*      CALCULATE INERTIA FORCES AT EACH DOF                           000021
00124    C                                                                      000021
00127          REWIND MASOFF                                                    000033
00130          DO 130 I=1,NDF                                                   000042
00133          FMOD(I) = 0.                                                     000044
00134          READ (MASOFF) (MASS(J), J=1,NDF)                                 000045
00142          DO 130 J=1,NDF                                                   000056
00145      130 FMOD(I) = FMOD(I) - MASS( J) *NODACC(J)                          000063
00145    C C                                                                    000063
00150          PRINT 3, (FMOD(I), I=1,NDF)                                      000072
00156        3 FORMAT ('0' , '   MODAL INERTIA LOADS TRANSFORM   FMOD = ',      000105
00156         *                 300(/15F8.2))                                   000105
00156    C                                                                      000105
00156    C                                                                      000105
00157          CALL TRANS(FMOD,NDF,TMDCK)                                       000106
00160          RETURN                                                           000112
00161          END                                                              000141
```

END OF COMPILATION: NO DIAGNOSTICS.

-127-

4.5 Description of Subroutine RBITRN

OUTLINE FOR SCIENTIFIC SUBPROGRAM DOCUMENTATION

IDENTIFICATION

Name/Title	SUBROUTINE RBITRN
Programmer/Date	FRED STRATMAN 4/74
Author/Date	FRED STRATMAN 5/74
Organization/Installation	LEC ASD FOR SMD
Machine Identification	UNIVAC 1110
Source Language	FORTRAN V

PURPOSE

RBITRN calculates an array of static forces and moments acting at the degrees of freedom of the loads model.

USAGE

- Calling Sequence

Call RBITRN

METHOD

- Symbol Definition

Coding	Type	Description	Units
A	A	Title on Flight Conditions File	
ACCEL	R	6 components of acceleration	in/sec^2, $rad./sec^2$
DMASS	R	Diagonal mass matrix	
FRB	R	Array of rigid-body forces and moments acting at the degrees-of-freedom of the loads model	
GTM	R	Mass matrix accounting for element location with respect to the c.g.	

Coding	Type	Description	Units
G	R	Computed gridpoint location vector matrix	
IDELMT	I	Element i.d. number from User Weight File	
IDXYZ	I	"node number * 10 + d.o.f."	
IFLTFL	I	Flight Conditions File unit number	
MASOFF	I	Unit having off diagonal mass matrices, if present	
MTAPE	I	Unit used for temporary storage of values used to calculate GTM	
NCID	I	Co-ordinate system i.d.	
NDF	I	Number of degrees of freedom	
NENG	I	Number of thrust components	
NEXT	I	Number of external force components	
NLINK	I	Number of link loads	
NMODES	I	Number of modes	
NNODES	I	Number of nodes	
NODEID	I	Node i.d.	
NSTART	I	Variable equal to number of data items to skip for binary reads	
OFFDIG	L	If true, off-diagonal elements in mass matrix	
QDD	R	Modal accelerations	
RMASS	R	Whole mass matrix = (NDF X NDF) for OFFDIG true	
RPMASS	R	The 6X6 mass matrix for each gridpoint from User Weight File	
SKIP SKIP	R	Locations into which irrelevant binary data is read	

-129-

RBITRN-2

Coding	Type	Description	Units
START STOP	R	Start and stop times of Flight Condition Data File	
TIME	R	Time of desired data from Flight Condition File	
TRBCK	H	Name passed to transform routine to verify that TRB matrix is read	
UMF	L	If true, read User Modal File If False, read User Weights File	
XCG YCG ZCG	R	Vehicle c.g. location	
XMAT	R	Locations of grid points in co-ordinate system	

- Model

The mass and gridpoint data for calculating rigid body inertia loads can be obtained from the User Modal File or the User Weights File.

If the User Modal data is used, the node i.d. numbers and locations and either the diagonal or whole mass matrix are simply read from the User Modal File described in Section 3.2.3.1. NODEID, XMAT, and DMASS, if present, are written on temporary storage unit MTAPE. If RMASS, the whole mass matrix, is present in place of DMASS, it is written on unit MASOFF.

The User Weights File has a format similar to the CONM1 and GRID cards which are NASTRAN bulk data cards. (See Section 3.2.3.2.1 and 3.2.3.2.2. The gridpoint locations are on the GRID cards while node i.d. numbers and a 6 X 6 mass matrix for each element are defined by CONM1 cards. Since IDXYZ's, the degree of freedom and gridpoint indicators, are not stored, they are computed. The (6 X 6) mass matrices read in for each node are mapped into an NDF X NDF mass matrix (where NDF = 6 * the number of gridpoints) as illustrated by Figure 4-3.

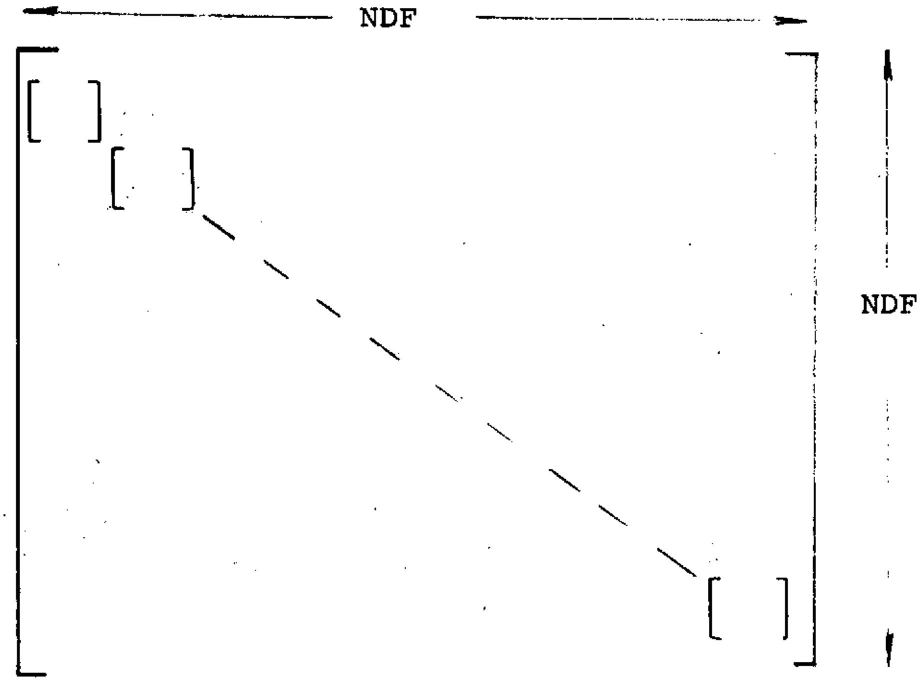


Figure 4-3
MASS MATRIX

If only a diagonal mass matrix is used the diagonal of length NDF is assigned to DMASS on unit MTAPE. If the whole mass matrix is used it is written by rows on unit MASOFF.

The steps followed in computing the array of rigid-body forces and moments on the loads model are:

1. Compute [G], the gridpoint location vector matrix, using the array of gridpoint locations [XMAT] and the X,Y,Z center-of-gravity locations.

2. Compute [GTM], the inertia matrix, by premultiplying [G] by the [MASS] matrix.

$$[GTM]_{(NDF \times 6)} = [MASS]\ [G]_{(6 \times NDF)}$$

   [MASS] can either be a vector of length NDF or if off-diagonal elements are present as illustrated in Figure 1, an NDF X NDF matrix.

3. Compute FRB by postmultiplying GTM by the 6 acceleration components of the c.g.

$$\{FRB\} = [GTM]\ \{-ACCEL\}$$

LISTING

A listing of subroutine RBITRN can be seen in section 4.5.1.

4.5.1 Listing of Subroutine RBITRN

```
0000 R 000205 FRB      0003   001626 FREQ     0000 R 000205 G        0003 R 002625 GTM      0000 I 000016 I
0000 I 000026 IBEGN    0000 I 000030 ID       0000 I 000023 IDELMT   0003 I 000032 IDXYZ    0000 I 000025 IEND1
0000 I 000027 IEND2    0000 I 000000 IFLTFL   0000 I 000022 IGRDPT   0000 I 000032 II       0000   005063 INJP$
0003 I 123527 IUMF     0000 I 000017 J        0000 I 000031 JP       0000 I 000020 K        0000 I 000010 LNK
0003 I 123530 MASOFF   0004 L 000001 MASOUT   0004 L 000002 MODLOT   0000 I 000001 MTAPE    0000 I 000024 NCID
0003 I 000030 NDF      0003   001625 NDFFE    0000 I 000034 NDIF     0000 I 000005 NENG     0000 I 000006 NEXT
0000 I 000007 NLINK    0003 I 000031 NMODES   0003 I 000664 NNODES   0000 I 000035 NODE     0003 I 000665 NODEID
0000 I 000033 NODEI    0000 I 000014 NSTART   0000 I 000021 NUM      0005   000015 NUMF     0005 L 000004 OFFDIG
0000 R 000041 ONE      0003   003457 PHI      0003 R 001627 QDD      0003   002625 RMASS    0003 R 002625 RPMASS
0000 R 000013 SKIP     0000 R 000205 SKP      0000 R 000011 START    0000 R 000012 STOP     0000 R 000015 T
0005 R 000006 TIME     0000 I 000002 TRBCK    0004 L 000000 TRNOUT   0005 L 000003 UMF      0005 R 000000 XCG
0000 R 000036 XDF      0003 R 001055 XMAT     0005 R 000001 YCG      0000 R 000037 YDF      0005 R 000002 ZCG
0000 R 000040 ZDF
```

```
00101    1*          SUBROUTINE RBITRN                                           000000
00101    2*     C                                                                000000
00101    3*     C* ROUTINE SETS UP THE RIGID-BODY-INERTIA LOADS TRANSFORMATION  000000
00101    4*     C                                                                000000
00103    5*          INCLUDE  DIM,LIST                                           000000
00103    5*     DIM  PROC                                                        000000
00103    5*     C*    ARRAY DIMENSIONS                                           000000
00103    5*     C                                                                000000
00104    5*          PARAMETER MAXDF = 410                                       000000
00105    5*          PARAMETER MAXMOD = 100                                      000000
00106    5*          PARAMETER MAXNOD = 120                                      000000
00106    5*     C                                                                000000
00106    5*      END                                                             000000
00107    6*          PARAMETER NDMAX = MAXDF                                     000000
00110    7*          DIMENSION G(6,NDMAX),GTM(NDMAX,6)                           000000
00111    8*          DIMENSION SKP(50)                                           000000
00112    9*           DIMENSION FRB(MAXDF)                                       000000
00113   10*           DIMENSION RPMASS(MAXNOD,4,6)                               000000
00114   11*          COMMON /MODAL/A(24),NDF,NMODES,IDXYZ(MAXDF),NNODES,NODEID(MAXNOD),   000000
00114   12*         *XMAT(MAXNOD,3),NDFFE,FREQ,QDD(MAXMOD),DMASS(MAXDF),RMASS(MAXDF),    000000
00114   13*         *PHI(MAXDF,MAXMOD),IUMF,MASOFF                       000000
00115   14*          COMMON /DATOUT/  TRNOUT,MASOUT,MODLOT                   000000
00116   15*          COMMON/TPINFO/XCG,YCG,ZCG,UMF,OFFDIG,FFREAD,TIME,ACCEL(6),NUMF   000000
00117   16*          EQUIVALENCE (FRB(1),SKP(1),G(1,1))                     NEW000000
00120   17*          EQUIVALENCE (RPMASS(1,1,1),RMASS(1),GTM(1,1))          NEW000000
00121   18*          LOGICAL FFREAD                                         -01000000
00122   19*          LOGICAL OFFDIG                                              000000
00123   20*          LOGICAL UMF                                                 000000
00124   21*          LOGICAL TRNOUT,MASOUT,MODLOT                                000000
00125   22*          DATA IFLTFL/4/                                              000000
00127   23*          DATA IUMF  /7/                                              000000
00131   24*          DATA MTAPE /8/                                              000000
00133   25*          DATA MASOFF/9/                                              000000
00135   26*          INTEGER TRBCK/6HTRB   /                                     000000
00137   27*          NAMELIST/ACC/ACCEL                                          000000
00137   28*     C                                                                000000
00140   29*          IF (.NOT. FFREAD) GO TO 32                                  000000
00140   30*     C                                                                000000
00140   31*     C*   READ VEHICLE ACCELERATIONS FROM THE FLIGHT CONDITIONS FILE  000000
00142   32*          REWIND  IFLTFL                                              000001
```

```
00143        READ (IFLTFL) A                                                       000004
00146        READ (IFLTFL) ENGID,EXTFOR,NENG,NEXT,NLINK,NNODES,LNK,START,STOP      000013
00146  C*    SKIP REST OF INFORMATION RECORDS                                      000013
00161        IF (NENG  .GT. 0) READ(IFLTFL) SKIP                                   000030
00165        IF (NEXT  .GT. 0) READ(IFLTFL) SKIP                                   000041
00171        IF (NNODES.GT. 0) READ(IFLTFL) SKIP                                   000052
00175        IF (NLINK .GT. 0) READ(IFLTFL) SKIP                                   000063
00201        NSTART = 6+NEXT+NENG                                                  000074
00202     27 READ(IFLTFL)        T, (SKP(I), I=1,NSTART) , ACCEL                   000101
00212        IF(NLINK .EQ. 0) READ(IFLTFL) (QDD(I),I=1,NNODES)                     000117
00221        IF(NLINK .NE. 0) READ(IFLTFL) (SKP(I),I=1,NLINK),                     000135
00221       *                              (QDD(I),I=1,NNODES)                     000135
00234     28 IF (T .LT. TIME) GO TO 27                                             000151
00236        WRITE (6,ACC)                                                         000166
00236  C                                                                           000166
00241     32 CONTINUE                                                              000173
00242        WRITE (6,99) (QDD(I),I=1,NNODES)                                      000173
00250     99 FORMAT (1H0, ' MODAL ACCELERATIONS ',10F10.4)                         000206
00251        IF (.NOT. UMF) GO TO 42                                               000206
00251  C*                                                                          000206
00251  C*    READ FROM USER MODAL FILE                                             000206
00251  C************************************************************************  000206
00253        CALL RDUMF                                                            000210
00254        WRITE(MTAPE) (IDXYZ(I),I=1,NDF)                                       000212
00254  C                                                                           000212
00262        IF (OFFDIG) GO TO 33                                                  000231
00262  C                                                                           000231
00264        WRITE(MTAPE) XCG,YCG,ZCG,(NODEID(J),J=1,NNODES),((XMAT(I,J),J=1,3)    000233
00264       (,  I=1,NNODES),(DMASS(K),K=1,NDF)                                     000233
00310        GO TO 63                                                              000266
00310  C                                                                           000266
00311     33 WRITE(MTAPE) XCG,YCG,ZCG,(NODEID(J),J=1,NNODES),((XMAT(I,J),J=1,3)    000270
00311       (,  I=1,NNODES)                                                        000270
00331        GO TO 63                                                              000320
00331  C                                                                           000320
00332     42 CONTINUE                                                              000322
00332  C                                                                           000322
00332  C*    READ FROM USER WEIGHTS FILE                                           000322
00333        READ (5,43) NNODES                                                    000322
00336     43 FORMAT (I3)                                                           000327
00336  C*                                                                          000327
00337        NDF = 6 * NNODES                                                      000327
00340        PRINT 307, NNODES,NDF                                                 000334
00340  C*                                                                          000334
00344        DO 144 J=1,NNODES                                                     000346
00347        NODEID(J) = J                                                         000355
00350        DO 144 I=1,6                                                          000362
00353        NUM = 6*(J-1) + I                                                     000362
00354    144 IDXYZ(NUM) = 10*J+I                                                   000365
00354  C*                                                                          000365
00354  C************************************************************************  000365
00354  C     READ GRIDPOINT ID*S AND THE MASS MATRIX                               000365
00354  C*     J = GRIDPOINT I.D.                                                   000365
00354  C*                                                                          000365
00357        DO 48 IGRDPT = 1,NNODES                                               000400
00362        READ (5,44,END=53) IDELNT, J, NCID, RPMASS(J,1,1),                    000400
00362       *RPMASS(J,2,1 ),RPMASS(J,2,2 ),RPMASS(J,3,1 ),RPMASS(J,3,2 ),IEND)     000400
```

```
00375          READ(5,45) IBEGN,RPMASS(J,3,3 ),RPMASS(J,4,1 ),RPMASS(J,4,2 ),       000421
00375         * RPMASS(J,4,3 ),RPMASS(J,4,4 ),RPMASS(J,5,1 ),RPMASS(J,5,2 ),        000421
00375         * RPMASS(J,5,3 ),IEND2                                                000421
00411          IF (IEND1 .NE.IBEGN) GO TO 64                                        000440
00413          READ(5,45) IBEGN,RPMASS(J,5,4 ),RPMASS(J,5,5 ),RPMASS(J,6,1 ),       000444
00413         * RPMASS(J,6,2 ),RPMASS(J,6,3 ),RPMASS(J,6,4 ),RPMASS(J,6,5 ),        000444
00413         * RPMASS(J,6,6)                                                       000444
00426          IF (IEND2 .NE.IBEGN) GO TO 64                                        000462
00430       48 CONTINUE                                                             000471
00430    C                                                                          000471
00432       53 CONTINUE                                                             000471
00432    C.                                                                         000471
00432    C.    READ GRID POINT LOCATIONS                                            000471
00433          DO 155 IGRDPT = 1,NNODES                                             000471
00436      155 READ (5,46,END=55) ID, (XMAT(ID,J), J=1,3)                           000475
00446       55 WRITE(MTAPE) (IDXYZ(I),I=1,NDF)                                      000517
00446    C                                                                          000517
00454          IF (OFFDIG) GO TO 54                                                 000534
00454    C                                                                          000534
00456          WRITE(MTAPE) XCG,YCG,ZCG,(NODEID(J),J=1,NNODES),((XMAT(I,J),J=1,3)   000536
00456         ,   I=1,NNODES),((RPMASS(J,K,K),K=1,6), J=1,NNODES)                   000536
00505          GO TO 63                                                             000576
00505    C                                                                          000576
00505    C.    WRITE OFF DIAGONAL ELEMENTS ONLY                                     000576
00506       54 CONTINUE                                                             000600
00507          WRITE(MTAPE) XCG,YCG,ZCG,(NODEID(J),J=1,NNODES),((XMAT(I,J),J=1,3)   000600
00507         ,   I=1,NNODES)                                                       000600
00507    CX                                                                  X      000600
00507    CX                                                                  X      000600
00527          DO 58 J=1,NNODES                                                     000637
00532          JP = 6*(J-1)                                                         000637
00532    C                                                                          000637
00533          DO 58 I=1,6                                                          000643
00533    C                                                                          000643
00536          DO 56 II = 1,NDF                                                     000651
00541       56 DMASS(II) = 0.                                                       000651
00543          DO 57 K=1,6                                                          000656
00546       57 DMASS(JP + K) = RPMASS(J,I,K)                                        000656
00550       58 WRITE(MASOFF) (DMASS(II), II=1,NDF)                                  000660
00560          GO TO 63                                                             000702
00560    C.                                                                         000702
00561       44 FORMAT (8X,3I8,5F8.2,I8)                                             000704
00562       45 FORMAT(I8,8F8.2,I8)                                                  000704
00563       46 FORMAT (8X,I8,8X,3F8.2)                                              000704
00564       47 FORMAT (1H0, ' DATA CARD OUT OF SEQUENCE NEAR ELEMENT ID NO.',I4)    000704
00565       64 PRINT 47 IDXYZ(J)                                                    000704
00570          STOP                                                                 000711
00570    C                                                                          000711
00570    C                                                                          000711
00571       63 CONTINUE                                                             000714
00571    C                                                                          000714
00572          REWIND  MTAPE                                                        000714
00573          READ (MTAPE) (IDXYZ(I),I=1,NDF)                                      000716
00601          WRITE (6,301) (IDXYZ(I), I=1,NDF)                                    000735
00607          IF (OFFDIG) GO TO 15                                                 000747
00607    C                                                                          000747
00611          READ (MTAPE) XCG,YCG,ZCG,(NODEID(J),J=1,NNODES),((XMAT(I,J),J=1,3)   000751
```

```
00611        ,+  I=1,NNODES),(DMASS(K),K=1,NDF)                                     000751
00635         GO TO 18                                                              001004
00636      15 READ (MTAPE) XCG,YCG,ZCG,(NODEID(J),J=1,NNODES),((XMAT(I,J),J=1,3)    001006
00636        (,  I=1,NNODES)                                                        001006
00636   C                                                                           001006
00656      18 CONTINUE                                                              001037
00657         WRITE (6,302)(NODEID(I),I=1,NNODES)                                   001037
00665         WRITE (6,303)((XMAT(I,J),J=1,3), I=1,NNODES)                          001056
00676         IF (MASOUT) PRINT 304,     (DMASS(K),K=1,NDF)                         001075
00705         DO 20 I=1,6                                                           001115
00710         DO 20 J=1,NDMAX                                                       001115
00713         G(I,J) = 0.                                                           001115
00714         GTM(J,I) =0.                                                          001115
00715      20 CONTINUE                                                              001123
00720         DO 130 J=1,NDF                                                        001123
00720   C.    COMPUTE NODE NUMBER                                                   001123
00723         NODEI=IDXYZ(J)/10                                                     001131
00723   C.    COMPUTE DEGREEE-OF-FREEDOM                                            001131
00724         NDIF= IDXYZ(J)-NODEI*10                                               001135
00725         K=0                                                                   001140
00726      80 CONTINUE                                                              001142
00727         K=K+1                                                                 001142
00730         IF (K .GT. NNODES) GO TO 160                                          001144
00732         IF(NODEID(K) .NE. NODEI) GO TO 80                                     001150
00734         NODE=K                                                                001155
00735         XDF=XMAT(NODE,1)-XCG                                                  001160
00736         YDF=XMAT(NODE,2)-YCG                                                  001163
00737         ZDF=XMAT(NODE,3)-ZCG                                                  001166
00740         ONE =1.                                                               001171
00741         GO TO (90,100,110,120,120,120) ,NDIF                                  001173
00742      90 CONTINUE                                                              001207
00743         G(1,J) = ONE                                                          001207
00744         G(5,J)= ZDF                                                           001210
00745         G(6,J)=-YDF                                                           001212
00746         GO TO 130                                                             001214
00747     100 CONTINUE                                                              001216
00750         G(2,J)=ONE                                                            001216
00751         G(4,J)=-ZDF                                                           001217
00752         G(6,J)=XDF                                                            001221
00753         GO TO 130                                                             001223
00754     110 CONTINUE                                                              001225
00755         G(3,J) =ONE                                                           001225
00756         G(4,J) =YDF                                                           001226
00757         G(5,J) =-XDF                                                          001230
00760         GO TO 130                                                             001232
00761     120 CONTINUE                                                              001234
00762         G(NDIF,J) = ONE                                                       001236
00763         GO TO 130                                                             001240
00764     160 PRINT 170                                                             001242
00766     170 FORMAT (1H1,10X, 'ERROR ENCOUNTERED IN GCALC ******', //)             001250
00767     130 CONTINUE                                                              001250
00767   C                                                                           001250
00771         IF ( .NOT. OFFDIG) GO TO 440                                          001250
00773         REWIND  MASOFF                                                        001252
00774         DO 420 J=1,NDF                                                        001255
00777         READ(MASOFF) (DMASS(K),K=1,NDF)                                       001266
01005         IF (MASOUT) PRINT 305, J, (DMASS(K),K=1,NDF)                          001277
```

```
01015          DO 420 I=1,6                                                    001316
01020          B= 0.0D0                                                        001324
01021          DO 410 K=1,NDF                                                  001332
01024          B=B+ DMASS(K)*G(I,K)                                            001332
01025      410 CONTINUE                                                        001336
01027          GTM(J,I) = B                                                    001336
01030      420 CONTINUE                                                        001346
01033          GO TO 450                                                       001346
01033   C                                                                      001346
01034      440 CONTINUE                                                        001352
01034   C                                                                      001352
01034   CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  001352
01034   CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  001352
01034   C                                                                      001352
01035          DO 250 I=1,6                                                    001352
01040          DO 250 J=1,NDF                                                  001373
01043          GTM(J,I)=DMASS(J) * G(I,J)                                      001373
01044      250 CONTINUE                                                        001401
01047      450 CONTINUE                                                        001401
01050          DO 300 I=1,NDF                                                  001401
01053          FRB(I) = 0.                                                     001407
01054          DO 300 J=1,6                                                    001412
01057      300 FRB(I) = FRB(I) - GTM(I,J) * ACCEL(J)                           001412
01062          PRINT 306, (FRB(I), I=1,NDF)                                    001424
01070          CALL TRANS(FRB,NDF,TRBCK)                                       001437
01071      301 FORMAT ('0',10X,'IDXYZS = ', 100(/20I6))                        001444
01072      302 FORMAT ('0',10X,'NODEIDS = ',100(/20I6))                        001444
01073      303 FORMAT ('0',10X,'XMAT = ',50(/4(3F8.2,4X)) )                    001444
01074      304 FORMAT ('0',10X,'DMASS = ', 25(/20F6.2))                        001444
01075      305 FORMAT ('0',10X,'OFFDIAGONAL MASS MATRIX, ROW NO. ' ,I4,25(/20F6.2  001444
01075         ,))                                                              001444
01076      306 FORMAT ('0', '    RIGID BODY INERTIA LOADS TRANSFORM  FRB = ',  001444
01076       *                    300(/15F8.2))                                 001444
01077      307 FORMAT ('0','  NNODES = ',I3/'    NDF = ',I3)                   001444
01100          RETURN                                                          001444
01101          END                                                             001501
```

END OF COMPILATION: NO DIAGNOSTICS.

4.6 Description of Subroutine RDUMF

OUTLINE FOR SCIENTIFIC SUBPROGRAM DOCUMENTATION

IDENTIFICATION

Name/Title	SUBROUTINE RDUMF
Programmer/Date	F. STRATMAN 4/74
Author/Date	F. STRATMAN 4/74
Organization/Installation	LEC-ASD FOR SMD
Machine Identification	UNIVAC 1110
Source Language	FORTRAN V

PURPOSE

RDUMF reads the User Modal File.

USAGE

- Calling Sequence

Call RDUMF

METHOD

RDUMF reads the User Modal File of the Format in Section 3.2.3.1. Unformatted read operations are used. The unit used for the file is IUMF. NUMF is the position of the desired file. If the first modal file is desired NUMF = 1, if the second NUMF = 2, etc... The value of NUMF is input through namelist $TPDAT. The MSC*LOCALIB File skipping routine, FSBSFL, is used to position the unit to the start of the desired data. A description of FSBSFL can be found in Appendix B. A flow chart of this subroutine can be found in Figure 4-4.

LISTING

A liting of subroutine RDUMF is seen in Section 4.6.1.

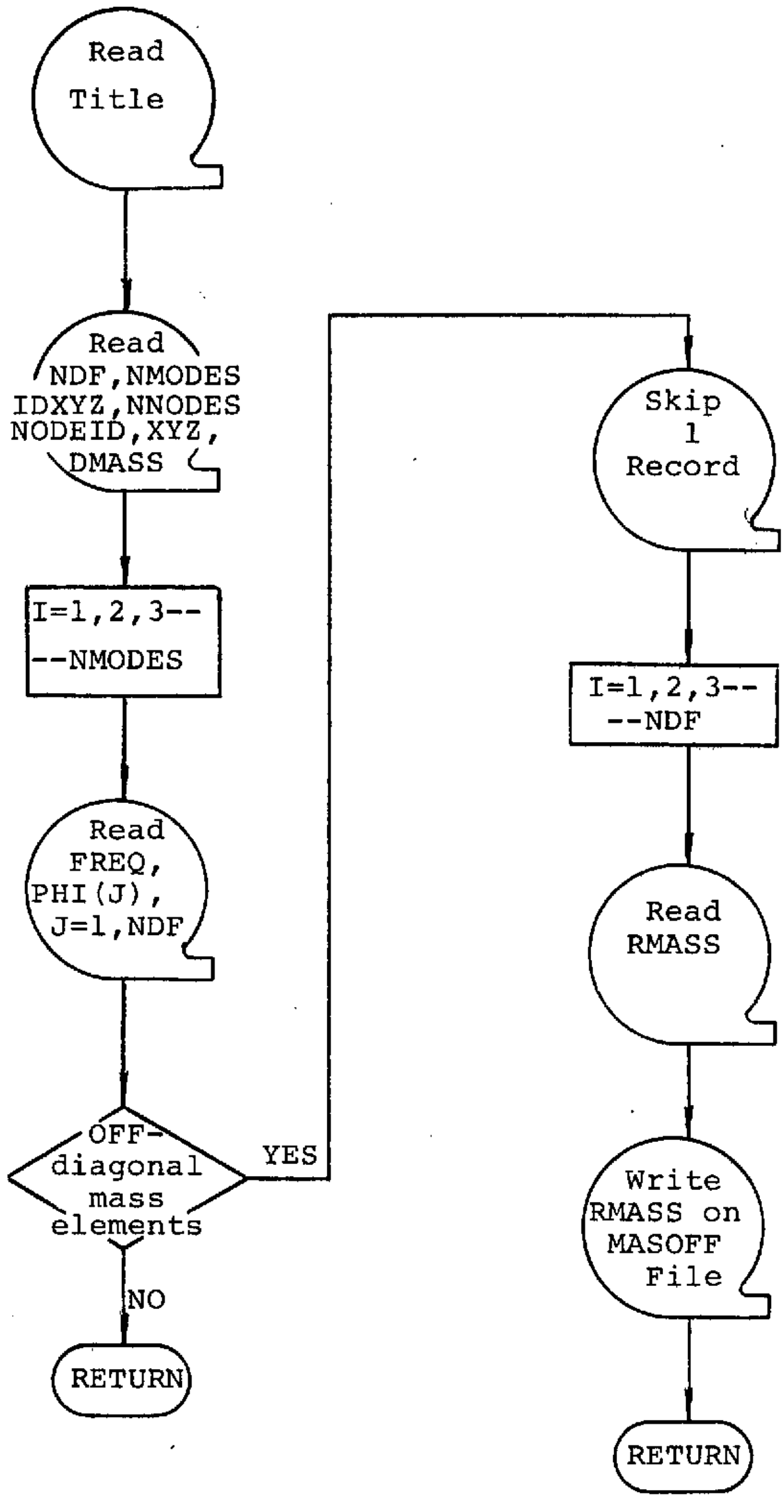


FIGURE 4-4

Flow Chart of Subroutine RDUMF

4.6.1 Listing of Subroutine RDUMF

```
@FOR,S   RDUMF,RDUMF,RDUMF
FOR S11X-12/23/79-21:45:25 (0,0)

     SUBROUTINE RDUMF       ENTRY POINT 000230

     STORAGE USED: CODE(1) 000242; DATA(0) 000073; BLANK COMMON(2) 000000

     COMMON BLOCKS:

      0003    MODAL  123531
      0004    TPINFO 000016
      0005    DATOUT 000003

     EXTERNAL REFERENCES (BLOCK, NAME)

      0006    FSBSFL
      0007    NRBUS
      0010    NIO1S
      0011    NIO2S
      0012    NPRTS
      0013    NWBUS
      0014    NERR3S

     STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

      0000    000005 101F      0000    000027 102F      0001    000012 121G      0001    000037 131G      0001    000051 136G
      0001    000056 142G      0001    000057 144G      0001    000070 151G      0001    000101 157G      0001    000114 167G
      0001    000122 174G      0001    000137 205G      0001    000164 220G      0001    000173 224G      0001    000204 232G
      0001    000211 SOL       0003 R  000000 A         0004    000007 ACCEL     0003 R  001773 DMASS     0004    000005 FFREAD
      0003 R  001626 FREQ      0000 I  000001 I         0003 I  000032 IDXYZ     0000    000047 INJP$     0000 I  000000 ISTATS
      0003 I  123527 IUMF      0000 I  000002 J         0000 I  000003 K         0003 I  123530 MASOFF    0005 L  000001 MASOUT
      0005 L  000002 MODLOT    0003 I  000030 NDF       0003    001625 NOFFE     0003 I  000031 NMODES    0003 I  000644 NNODES
      0003 I  000665 NODEID    0004 I  000015 NUMF      0004 L  000004 OFFDIG    0003 R  003457 PHI       0003    001627 QDD
      0003 R  002625 RMASS     0000 R  000004 SKIP      0004    000006 TIME      0005 L  000000 TRNOUT    0004    000003 UMF
      0004    000000 XCG       0003 R  001055 XMAT      0004    000001 YCG       0004    000002 ZCG


 00101      1*              SUBROUTINE RDUMF                                                     000000
 00101      2*       C      A ROUTINE TO READ THE USER MODAL FILE                                000000
 00103      3*              LOGICAL OFFDIG                                                       000000
 00104      4*              INCLUDE  DIM,LIST                                                    000000
 00104      4*       DIM  PROC                                                                   000000
 00104      4*       C*     ARRAY DIMENSIONS                                                     000000
 00104      4*       C                                                                           000000
 00105      4*              PARAMETER MAXDF = 410                                                000000
 00106      4*              PARAMETER MAXMOD = 100                                               000000
 00107      4*              PARAMETER MAXNOD = 120                                               000000
 00107      4*       C                                                                           000000
 00107      4*        END                                                                        000000
 00110      5*              COMMON /MODAL/A(29),NDF,NMODES,IDXYZ(MAXDF),NNODES,NODEID(MAXNOD),   000000
 00110      6*             *XMAT(MAXNOD,3),NOFFE,FREQ,QDD(MAXMOD),DMASS(MAXDF),RMASS(MAXDF),     000000
```

```
00110        *PHI(MAXDF,MAXMOD),IUMF,MASOFF                                  000000
00111         COMMON/TPINFO/XCG,YCG,ZCG,UMF,OFFDIG,FFREAD,TIME,ACCEL(6),NUMF 000000
00112         COMMON /DATOUT/  TRNOUT,MASOUT,MODLOT                          000000
00113         LOGICAL TRNOUT,MASOUT,MODLOT                                   000000
00114         DATA IUMF  /7/                                                 000000
00114   C.    SKIP TO DESIRED DATA                                        NEW000000
00116         CALL FSBSFL (IUMF,NUMF,ISTATS)                              NEW000000
00117         READ(IUMF) (A(I),I=1,12)                                       000004
00125         READ(IUMF) NDF,NNODES,(IDXYZ(I), I=1,NDF),NNODES,(NODEID(J),J=1,  000015
00125        $   NNODES),((XMAT(I,J),J=1,3),I=1,NNODES),(DMASS(K),K=1,NDF)   000015
00155         PRINT 101, (A(I), I=1,12), NDF,NNODES, NNODES               -02000073
00166         DO 32 I=1,NNODES                                               000114
00171         READ(IUMF) FREQ,(PHI(I,J),J=1,NDF)                             000114
00200      32 IF(MODLOT) PRINT 102, I,FREQ,(PHI(I,J), J=1,NDF)               000125
00212         IF (.NOT. OFFDIG) GO TO 50                                     000155
00214         READ(IUMF) SKIP                                                000157
00217         DO 35 I=1,NDF                                                  000166
00222         READ (IUMF) (RMASS(J) , J=1,NDF)                               000166
00230      35 WRITE(MASOFF) (RMASS(J) , J=1,NDF)                             000176
00237      50 RETURN                                                         000211
00240     101 FORMAT ('0', 20A6/10X,'NO. DEGREES-OF-FREEDOM = ',I4/10X,'NO. OF  000241
00240        *MODES = ',I4/10X,'NO. OF NODES = ',I4)                         000241
00241     102 FORMAT ('0', ' MODE NO. ', I3,'  ,FREQ. = ', F9.4,' -PHIS = ',  000241
00241        *                                            10(/10F10.2))     000241
00242         END                                                            000241

   END OF COMPILATION:        NO  DIAGNOSTICS.
```

-140-

4.7 Description of Subroutine TRANS

OUTLINE FOR SCIENTIFIC SUBPROGRAM DOCUMENTATION

IDENTIFICATION

Name/Title	SUBROUTINE TRANS
Programmer/Date	F. STRATMAN 4/74
Author/Date	F. STRATMAN 5/74
Organization/Installation	LEC-ASD FOR SMD
Machine Identification	UNIVAC 1110
Source Language	FORTRAN V

PURPOSE

TRANS premultiplies the force and moment arrays setup in RBITRN, MILTRN, and AFTRAN by their corresponding "loads model to finite element" transform matrices, punchs NASTRAN "Force" and "moment" cards, and writes NASTRAN force card images on the NASTRAN Input Loads File.

USAGE

- Calling Sequence

    Call TRANS (POSTML, NCOLS, ARAYCK)

Parameter	Dimension	Type	Description
POSTML	MAXDF	R	The array of forces and moments acting at the degrees of-freedom of the loads model
NCOLS		I	The number of columns filled in POSTML
ARAYCK		H	1-word identifier of POSTML

- Error Messages

When the wrong transform array card deck is detected upon comparing ARAYCK with the data deck identifier, the following message is printed. DATA DECK OUT OF SEQUENCE

METHOD

- Symbol Definition

Coding	Type	Description	Units
ASSIGN	R	(in text)	
BUFF	A	A 280 word buffer to store the NASTRAN force card images as they are computed	
CID	I	Coordinate system i.d.	
DATNAM	A	The one word i.d. of the transform data deck	
FFEVAL	R	The finite element force value being computed	
IPOS	I	Variable used to identify the position in an array of a certain desired value	
IROW	I	The degree-of-freedom of the finite element model for which a force is being computed	
ISTART	I	Position in BUFF where data encoding is to start	
NAME	A	Identifies "FORCE" or "MOMENT" NASTRAN card	
NASFIL	I	Unit number of NASTRAN INPUT LOADS File	
NDFFE	I	Number of degrees of freedom in the finite element model	
PREMUL	R	Premultiplier - the transform matrix	
SCALEF	R	Scale factor - CMASS for force card, CINERT for moment card	

Coding	Type	Description	Units
SID	R	Load set i.d. number	
VECTOR	R	3-component force vector for NASTRAN card	

- Model

The general transform process is described in Section 2.1. Due to the limited core space available only one finite element force value is processed at a time.

The processing consists of:

1. reading a row of the transform matrix
2. postmultiplying by the array of forces on the loads model
3. creating the NASTRAN Force Card image
4. writing the card image on NASFIL and punching a NASTRAN bulk data card.

The NASFIL format is illustrated in Section 3.1.5. The NASTRAN bulk data "FORCE" card is illustrated in Section 3.3.

ASSIGN is an NDFFE X 3 matrix which corresponds to the transform matrix. For each non-zero row of the transform matrix there is a corresponding row of ASSIGN which provides the following information about the corresponding FFE value computed:

Column	
1.	finite element gridpoint i.d. number
2.	1 → static force being created 2 → static moment being created
3.	1,2,3 corresponding to X,Y,Z d.o.f. associated with the gridpoint

The program flow is shown in Figure 4-5.

LISTING

A listing of subroutine TRANS is found in Section 4.7.1.

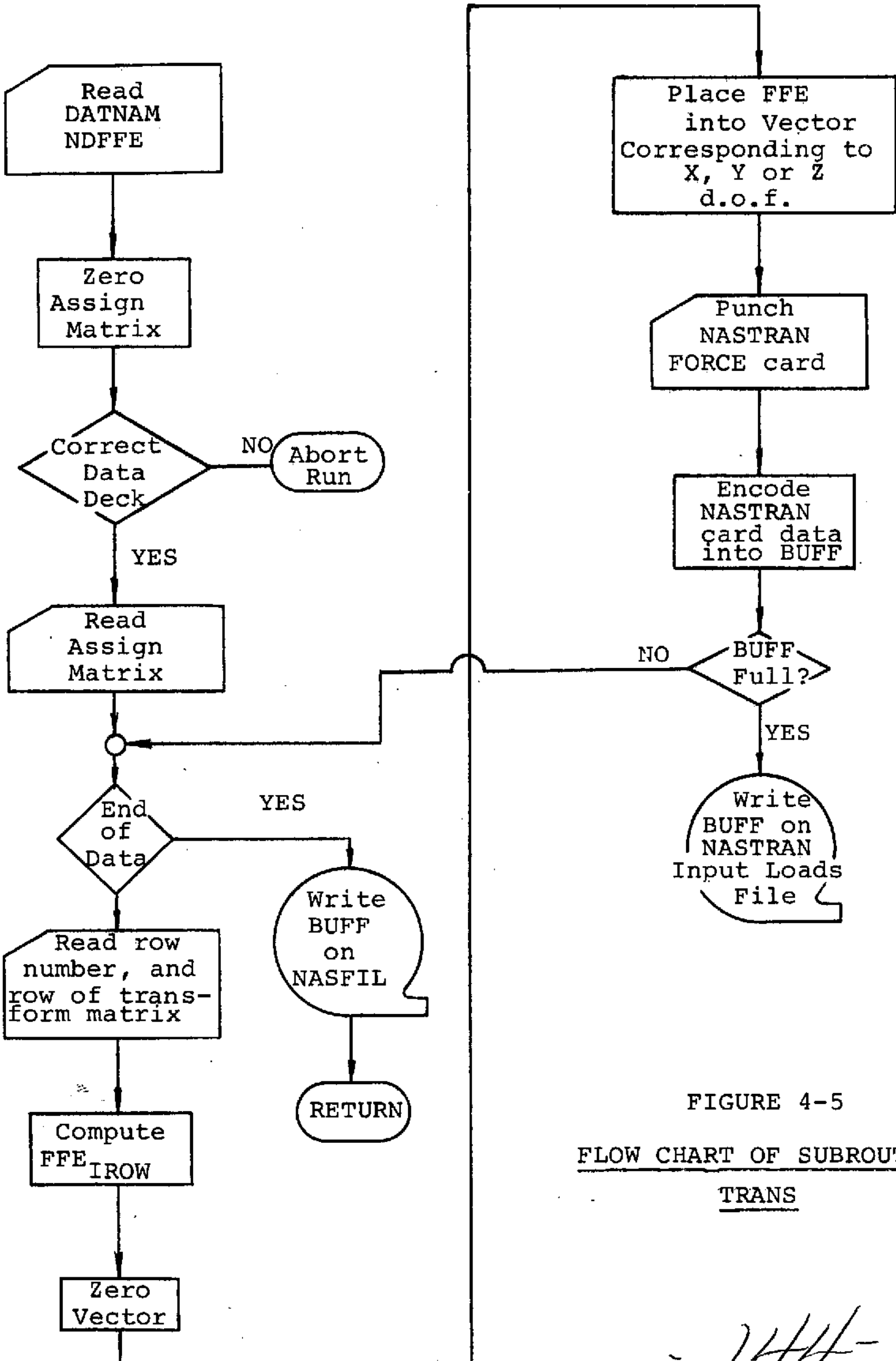


FIGURE 4-5

FLOW CHART OF SUBROUTINE TRANS

-144-

4.7.1 Listing of Subroutine TRANS

```
@FOR,S   TRANS,TRANS
FOR SEIX-12/23/79-21:45:30 (0,)

   SUBROUTINE TRANS      ENTRY POINT 000323

   STORAGE USED: CODE(1) 000336; DATA(0) 001443; BLANK COMMON(2) 000000

   COMMON BLOCKS:

    0003    MODAL  123531
    0004    FACTOR 000004
    0005    DATOUT 000003

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006    NTRAN
    0007    NNCODS
    0010    NRNLS
    0011    NPRTS
    0012    NIO2S
    0013    NSTOPS
    0014    NIO1S
    0015    NWDUS
    0016    NIO3S
    0017    NERR3S

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000015 145G      0001   000016 150G      0001   000046 167G      0001   000051 172G      0001   000073 206G
    0001   000104 214G      0001   000115 223G      0001   000042 305L      0000   000524 325F      0000   000530 326F
    0000   000533 328F      0000   000534 329F      0001   000236 330L      0000   000542 345F      0000   000544 346F
    0001   000254 350L      0000   000545 351F      0003   000000 A         0000   000506 ARR       0000   000465 ASSGN
    0003 I 003457 ASSGN1    0003 I 003457 ASSGN2    0003 I 003457 ASSGN3    0003 I 003457 ASSIGN    0000 R 000000 BUFF
    0004 L 000003 CARDS     0000 I 000444 CID       0004   000001 CINERT    0004   000000 CMASS     0000 I 000443 DATNAM
    0003   001773 DMASS     0000 R 000453 FFEVAL    0003   001626 FREQ      0000 I 000451 I         0000 I 000432 IBLANK
    0003   000032 IDXYZ     0000 I 000454 II        0000   001424 INJP$     0000 I 000455 IPOS      0000 I 000447 IROW
    0000 I 000450 ISTART    0003   123527 IUMF      0000 I 000452 J         0000 I 000456 L         0003   123530 MASOFF
    0005 L 000001 MASOUT    0005 L 000002 MODLOT    0000 I 000430 NAME      0000 I 000445 NASFII    0003   000030 NDF
    0003 I 001625 NDFFE     0000 I 000446 NINE      0003   000031 NMODES    0003   000664 NNODES    0003   000665 NODEID
    0003   003457 PHI       0000 R 000556 PREMUL    0003   001627 WDD       0003   002625 RMASS     0004 R 000000 SCALEF
    0004 I 000002 SID       0000 R 000556 TAP       0000 R 000556 TMOD      0000 R 000556 TRH       0005 L 000000 TRNOUT
    0000 R 000440 VECTOR    0003   001055 XMAT

00101             SUBROUTINE TRANS(POSTML,NCOLS,ARAYCK)                                      000001
00101        C                                                                               000001
00101        C*   THIS ROUTINE READS ONE ROW OF A PREMULTIPIER AT A TIME, MULTIPLIES         000001
00101        C*  BY THE POSTMULTIPLIER IF NECESSARY, AND PRINTS THE NASTRAN FORCE OR         000001
00101        C*  OR MOMENT CARD                                                              000001
00101        C                                                                               000001
00103             INCLUDE  DIM,LIST                                                          000001
```

```
00103        DIM  PROC                                                          000001
00103        C*     ARRAY DIMENSIONS                                            000001
00103        C                                                                  000001
00104              PARAMETER MAXDF = 410                                        000001
00105              PARAMETER MAXMOD = 100                                       000001
00106              PARAMETER MAXNOD = 120                                       000001
00106        C                                                                  000001
00106         END                                                               000001
00107              DIMENSION BUFF(280)                                          000001
00110              DIMENSION POSTML( MAXDF),PREMUL(MAXDF)                       000001
00111              DIMENSION NAME(2),IBLANK(6),SCALEF(2),VECTOR(3)              000001
00112              DIMENSION TRB(MAXDF),TAP(MAXDF),THOD(MAXDF)                  000001
00113              COMMON /MODAL/A(24),NDF,NMODES,IDXYZ(MAXDF),NNODES,NODEID(MAXNOD),   000001
00113             *XMAT(MAXNOD,3),NDFFE,FREQ,QDD(MAXMOD),DMASS(MAXDF),RMASS(MAXDF),    000001
00113             *PHI(MAXDF,MAXMOD),IUMF,MASOFF                                000001
00114              COMMON/FACTOR/CMASS,CINERT,SID,CARDS                         000001
00115              COMMON /DATOUT/  TRNOUT,MASOUT,MODLOT                        000001
00116              LOGICAL TRNOUT,MASOUT,MODLOT                                 000001
00117              INTEGER ARAYCK,DATNAM,ASSIGN(MAXDF,3),ASSGN1(MAXDF,3),       000001
00117             *          ASSGN2(MAXDF,3),ASSGN3(MAXDF,3)                    000001
00120              INTEGER  CID/0/                                              000001
00122              INTEGER SID                                                  000001
00123              LOGICAL CARDS                                                000001
00124              DATA NASFIL/10/                                              000001
00126              DATA NINE/6H999999/                                          000001
00130              DATA NAME/6HFORCE ,6HMOMENT/                                 000001
00132              DATA IBLANK/6*6H      /                                      000001
00134              EQUIVALENCE  (SCALEF(1),CMASS)                               000001
00135              EQUIVALENCE  (SCALEF(2),CINERT)                              000001
00136              EQUIVALENCE (PREMUL(1),TRB(1),TAP(1),THOD(1))                000001
00137              EQUIVALENCE (ASSIGN(1,1),ASSGN1(1,1),ASSGN2(1,1),ASSGN3(1,1),   000001
00137             *             PHI(1,1))                                       000001
00140              NAMELIST/ASSGN/DATNAM,NDFFE,ASSGN1,ASSGN2,ASSGN3             000001
00141              NAMELIST/ARR/TRB,TAP,THOD,IROW                               000001
00142              CALL NTRAN (NASFIL,10)                                       000002
00143              ISTART = -13                                                 000006
00144              DO 300 I=1,NDFFE                                             000016
00147              DO 300 J=1,3                                                 000016
00152          300 ASSIGN(I,J) = 0                                              000016
00155              READ(5,ASSGN)                                                000023
00160              IF (DATNAM .EQ. ARAYCK) GO TO 305                            000027
00162              PRINT 329                                                    000033
00164              STOP                                                         000037
00165          305 CONTINUE                                                     000042
00166              DO 330 I=1,NDFFE                                             000042
00171              DO 310 J=1,NCOLS                                             000051
00174          310 PREMUL(J) = 0.                                               000051
00176              READ (5,ARR,END=350)                                         000052
00201              IF (TRNOUT) PRINT 351, DATNAM,IROW, (PREMUL(J), J=1,NCOLS)   000061
00212              FFEVAL  = 0.                                                 000076
00213              DO 320 J=1,NCOLS                                             000104
00216          320 FFEVAL  = FFEVAL  + PREMUL(J) *POSTML(J)                     000104
00220  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00220              IF (FFEVAL .EQ. 0.) GO TO 330                                000110
00222              DO 323 II = 1,3                                              000115
00225          323 VECTOR(II) = 0.                                              000115
00227              IPOS = ASSIGN(IROW,3)                                        000116
```

```
00230              VECTOR(IPOS) = FFEVAL                                                000122
00231              IPOS = ASSIGN(IROW,2)                                                000124
00231        C*    WRITE OR PUNCH NASTRAN CARD                                          000124
00232              IF(CARDS) WRITE(31,326) NAME(IPOS),SID,ASSIGN(IROW,1),CID,           000130
00232             *                        SCALEF(IPOS),VECTOR                          000130
00243              WRITE (6,325)           NAME(IPOS),SID,ASSIGN(IROW,1),CID,           000153
00243             *                        SCALEF(IPOS),VECTOR                          000153
00253              ISTART = ISTART + 14                                                 000174
00254              IF (ISTART .EQ. 281) ISTART = 1                                      000177
00256              ENCODE (84,328,BUFF(ISTART)) NAME(IPOS),SID,ASSIGN(IROW,1),CID,      000205
00256             *                        SCALEF(IPOS),VECTOR,IBLANK                   000205
00256        C*    IF FULL, WRITE BUFFER ON NASTRAN INPUT LOADS FILE                    000205
00267          330 IF(ISTART .EQ. 267) CALL NTRAN(NASFIL,1,280,BUFF,L,22)               000236
00267        C                                                                          000236
00267        C*    WRITE CONTENTS OF BUFFER AND END INDICATORS ON NASTRAN INPUT FILE    000236
00272          350 IF (ISTART .EQ. 267) ISTART = -13                                    000254
00274              ISTART = ISTART + 14                                                 000261
00275              ENCODE (6,328,BUFF(ISTART)) NINE                                     000264
00300              CALL NTRAN(NASFIL,1,ISTART,BUFF,L,22)                                000276
00301          325 FORMAT(1HO,A6,2X,3I8,4F8.2)                                          000306
00302          326 FORMAT   (A6,2X,3I8,4F8.2)                                           000306
00303          328 FORMAT (14A6)                                                        000306
00304          329 FORMAT ( 'DATA DECK OUT OF SEQUENCE')                                000306
00305          345 FORMAT ( 11F7.2)                                                     000306
00306          346 FORMAT (3I1)                                                         000306
00307          351 FORMAT ('0',A6,' MATRIX, ROW NUMBER',I4, 50(/10F10.2))               000306
00310              RETURN                                                               000306
00311              END                                                                  000335

    END OF COMPILATION:          1 DIAGNOSTICS.
```

5.0 REFERENCES

1. Oney, J. K., "Detailed Requirements Document for the Integrated Structural Analysis System (Phase A), Lockheed Electronics Document 0869, January, 1974.

2. Rainey, J. A., "Functional Design Specifications Document for the Integrated Structural Analysis System (ISAS) Design (Phase A), Lockheed Electronics Document 1808, June 1974.

3. Schwartz, R. D., "Summary of ISAS Program Requirements", Structures Branch Report 72-ES2-14, NASA/Lyndon B. Johnson Space Center, Houston, Texas, December, 1972.

4. The NASTRAN Users Manual, Mc Cormick, C. W., ed., NASA SP-222, October, 1969.

APPENDIX A

NTRAN ROUTINE DESCRIPTION

- Description

If the file is on tape, no action is taken. If the file is on drum, the relative drum address is adjusted by the word count in BFLNG. If the file is on FASTRAND, the relative sector address is adjusted by the word count in BFLNG divided by the sector size. The adjustments are positive, unless the function is backspace, in which case it is negative. Before exiting, zero is stored in BFLNG, so that a redundant call on UPDDA$ will not destroy the mass storage location.

4.17.21. NWBLK$ – WRITE BUFFERED OUTPUT BLOCK TO MASS STORAGE

- Purpose

To output a block of nonformatted or formatted data to tape, drum, or FASTRAND.

- FORTRAN V Reference

Not applicable.

- Assembler Language Reference

There are two entries in NWBLK$: DRAIN$ and WRBLK$. DRAIN$ is invoked by

```
SLJ                     DRAIN$
```

On entry, UNIT$ must contain the unit number and A0 must contain the packet location.

WRBLK$ is invoked by,

```
SLJ                     WRBLK$
```

On entry, A0 must contain the location of the I/O packet.

- Description

NWBLK$ has two entries, DRAIN$ and WRBLK$. DRAIN$ inserts any trail pad needed by the block. For nonformatted output DRAIN$ also inserts the checksum and control words. The last control word is updated and WRBLK$ is called to write the data block. For nonformatted blocks the checksum is then set to zero and the block sequence number is incremented.

WRBLK$ waits until the previous operation on the file is completed. It then sets the function code and calls UPDDA$ to update the mass storage location. The buffer location and data length are inserted into the packet and, if two buffers are in use, the old buffer is inserted as the available or working buffer. Finally IO$ is called to write the block. WRBLK$ does not wait for the write to be completed before exiting.

4.18. NTRAN I/O PROCESSING

The READ and the WRITE statements, although convenient to use, do not make efficient use of tapes and drums because the language does not permit parallel processing. Furthermore, a considerable amount of time is used in processing an I/O list because of its generality.

NTRAN provides a tool for reading and/or writing binary information on tape or drum. It also provides I/O buffering through a CALL statement in the FORTRAN language:

CALL NTRAN (UNIT, sequence of operations)

in which UNIT is an integer constant or variable designating the logical unit. The sequence of operations is any list of I/O operations (as specified in 4.18.1) to be performed in order on the specified unit. If the unit is not busy, NTRAN initiates the first operation and stacks the rest in a waiting list and then returns to the calling program. If the unit is already busy, then the entire sequence is stacked in a waiting list and chained to any previously stacked operations. The exceptions are operations 16 to 22; when they are encountered, NTRAN waits for the completions of all previous operations for that unit before returning to the calling program.

When an interrupt occurs, NTRAN records the transmission status, initiates the next operation in the chain, and returns control to the interrupted calling program.

Input/output operations provided by NTRAN are:

(1) Write (tape, drum, or FASTRAND)

(2) Read (tape, drum, or FASTRAND)

(3) Block Read (tape, drum, or FASTRAND)

(4) Search Read (tape, drum, or FASTRAND)

(5) Search Drum (drum or FASTRAND)

(6) Position Drum (drum or FASTRAND)

(7) Position Tape by Block (tape only)

(8) Position Tape by Files (tape only)

(9) Write End of File (tape only)

(10) Rewind (tape, drum, or FASTRAND)

(11) Rewind/Interlock (same as 10 for drum and FASTRAND)

(12) Set Tape Density Medium (tape only)

(13) Set Tape Density Low (tape only)

(14) Set Tape Parity Odd (tape only)

(15) Set Tape Parity Even (tape only)

(16) Initialize Multireel File (tape only)

(17) Swap Reels for Multireel File (tape only)

(18) Reassign Unit (tape, drum, and FASTRAND)

(19) Assign Unit to External File Name (tape, drum, or FASTRAND)

(20) NOP (tape, drum, and FASTRAND)

(21) Get device

(22) Wait and Unstack then Release Unit (tape, drum, or FASTRAND)

(23) Set Tape Density High (tape only)

In order to use NTRAN, a FORTRAN program must have some way to check the status of the transmission. For this reason, every block of main storage which is used for I/O has a block status word (an integer variable) associated with it; the name of the status word is specified in the argument list of the CALL.

When NTRAN is called, the list of arguments is searched for status words, and these are all set to a value (−1) which indicates transmission-not-complete. When an interrupt occurs, the corresponding status word is set by NTRAN to a value which indicates the nature of completion, whether normal (a positive value indicating number of words transmitted), abnormal (value = −2) or in error (value = −3 or −4). The status words for each operation are defined in 4.18.1.

When NTRAN generates −2 or −3, it releases all operations stacked for the unit which have not been started. The offending operation is marked abort and is left stacked. Any further calls of NTRAN, requesting the above described unit (except operation 22) will not be performed or stacked, but will generate a particular status code (−4). Operation 22 may be used to release the abort condition for a unit. This allows the programmer to regain control after trying to read or write past an end-of-file, end-of-drum, or end-of-tape.

An attempt to read or write zero words (N=0) will result in the function being ignored.

The following errors will generate a status word of −3:

(1) Hardware errors.

(2) Parity and character count errors.

(3) Illegal unit specified.

NOTE: Legal units are all tapes and drum/FASTRAND files.

4.18.1. OPERATIONS

An operation is defined in the argument list by a group of arguments. The first argument for an operation identifies the type of operation. It is followed by the parameters for the operation; these are fixed in number and order of occurrence by the type of operation. Several operations may be grouped in a single call to NTRAN.

When referencing a drum file, the current drum address for that file is the starting address for the file only if the drum file was never referenced in the current run. If the drum file was referenced before, the current drum address is the current address before the last CALL of the file plus the number of words transmitted or positioned in that CALL. In order to reach the starting address of the file, operation 10 and 22 can be used.

For example, in the call NTRAN (3, 9, 10, 22):

3 = unit number

9 = end-of-file when operation is completed

10 = rewind unit

22 = all operations on unit must be completed before another function is issued.

The cited example is a stacked operation.

NOTE: For FASTRAND I/O the specified drum address is a sector count and not a word count as for drum I/O. However, with normal termination, the status variable associated with a main storage transfer will indicate actual number of words transmitted. It is then up to the user to perform the covered divide with the sector size in order to retrieve the corresponding sector count.

For search operations on FASTRAND, if a find is made, the drum address will point to the sector containing the matching item; a following read function will therefore not necessarily start reading the matched item.

(a) Write

The argument group is: 1, N, B, L

in which N is an integer constant or variable which specifies the block length; B is a variable name from which data is to be written; and L is an integer variable, the status word, which is set by NTRAN as follows:

−1 = transmission not complete

−2 = end-of-tape or drum file

−3 = device error

−4 = transmission aborted (previous operation had −2 or −3 status).

If the transmission is completed normally, L receives the number of words transmitted (N).

(b) Read

The argument group is: 2, N, B, L

in which N is an integer constant or variable which specifies the length of the main storage block which will receive the data (for tape, N is the maximum number of words which will be transmitted from the tape block; for drum, N words will be transmitted), B is a variable which is the name of a main storage area into which the data is to be read, and L is an integer variable (the status word), which is to be set as follows:

−1 = transmission not complete

−2 = end of file (no words read DRUM)

−3 = device error

−4 = transmission aborted (previous operation had −2 or −3 status)

If the transmission is completed normally, L receives the number of words transmitted (N).

(c) Block Read

The argument group is: 3, N, B, L

A block read for tape and FASTRAND is the same as an ordinary read. For drum, transmission is terminated by reading a word of all 1 bits (called end-of-block word). N is the maximum number of words which can be transmitted. L (the status variable) receives the actual number of words transmitted if the operation is completed normally; otherwise L is set as in READ. B has the same definition as in read.

(d) Search Read

The argument group is: 4, S, N, B, L

in which S (a sentinel word) is a constant or a variable which is used in searching tape or drum.

For tape, the first word of each block is compared to the sentinel and, when a match is found, that block (including the sentinel word) is read. For drum, starting at the current drum address, each word is compared to the sentinel and, when a match is found, that block (N words) is read. An unsuccessful search results in an end-of-file status (−2) for L. For FASTRAND a track search is employed; if no find is made, the user may request additional searches.

When a match is found on FASTRAND, the entire sector containing the matched sentinel will be read into B.

(e) Drum Search

The argument group is: 5, S

in which S is a constant or variable sentinel word. Starting at the current drum address, the drum is searched until match or end-of-file is reached. The drum address of the match becomes the new current drum address (first drum address to be read or written is that of matched drum address). When a match is found on FASTRAND, the drum address will point to the sector containing the matched sentinel. If a match has not been made, the address does not change.

(f) Position Drum

The argument group is: 6, N

in which N is an integer constant or variable, positive or negative, which is added to the current drum address to form a new current drum address. If N is negative and the current drum address plus N is less than the starting address of drum file, current drum address is set to starting address of drum file. N is word count for drum, and sector count for FASTRAND.

(g) Position Tape By Blocks

The argument group is: 7, N

in which N is an integer constant or variable which specifies the number of blocks to space over on tape. Positive N for forward spacing; negative N for backspacing.

(h) Position Tape By Files

The argument group is: 8, N

in which N is an integer constant or variable which specified the number of file marks to space over. Positive N for forward spacing; negative N for backspacing. The operation is terminated by moving over the Nth file mark, by reaching the load point (back spacing), or by reaching the end of tape (forward spacing).

(i) End File

The argument group is: 9

For tape an end-of-file mark is written.

(j) Rewind

The argument group is: 10

(k) Rewind/Interlock

The argument group is: 11

For tape, a rewind/interlock is given. For drum and FASTRAND the operation is the same as a rewind.

NOTE: Operations l through o pertain to magnetic tape density and parity setting (are available only on UNISERVO IV-C, VI-C, and VIII-C units). If not specified, the setting will be system standard.

(l) Set Tape Density Medium (556 bpi)

The argument group is: 12

(m) Set Tape Density Low (200 bpi)

The argument group is: 13

(n) Set Tape Parity Odd (binary standard)

The argument group is: 14

(o) Set Tape Parity Even (BCD standard)

The argument group is: 15

NOTE: Density and parity setting routines set density and parity for all tape units tied to a logical unit when multireel processing.

If the I/O error routine has been entered by a call to NERCR$ and an ERR clause has been found, control goes to the statement number specified in the FORTRAN I/O reference. Control goes back to the I/O routine which detected the error, if entry point NERCR$ was used and no ERR clause has been specified. In this case, the following identifying message is printed:

'I/O CALLED AT SEQUENCE NUMBER $ OF P'

where $ is the sequence number shown in the compiler listing and P stands for MAIN PROGRAM or the name of the subprogram.

If the I/O error routine is entered by a call to NERCT$ and an ERR clause has been specified, control changes to the I/O calling program without the identifying message. If NERCT$ is called and no ERR clause has been specified, the message

'FORTRAN V ERROR TERMINATION:'

is printed by NERR6$. This is followed by a printed line identical to the identifying message. The ensuing walk back process prints one line for each subprogram over which walk back occurs indicating sequence or line number and the name of the subroutine. Walk back ceases when the main program is reached or when 200 lines have been printed.

The I/O error routine is entered at NEFCL$ when a software end of file or @EOF card or the image @EOF+x (x is nonsignificant) as the first word of a record is encountered. If an END clause has been specified, control goes to the I/O calling program without any messages at all. If no END clause is specified, the following message is printed:

'EXECUTION TERMINATED BY AN ATTEMPT TO READ PAST AN END OF FILE.'

This is followed by the walk back process as described previously.

4.19.1. ERROR CODES AND ASSOCIATED MESSAGES

The following is a list of error codes, the contents of the FORTRAN V error word (NSTAT$), and the error message provided by the I/O routine where the error was detected.

00	FORMAT Type	I/O List Type	Location of Variable in Program

Remark: Type of FORMAT specifications does not agree with type specified by I/O list.

→ Message: 'ERROR – TYPE IN FORMAT IS NOT THE SAME AS THE INTERNAL TYPE.'

Record image is displayed. Message is printed no more than once if error recurs in same I/O statement. No termination, if ERR clause is not specified.

01	Character	00	Location of Word in Format

Message: 'ILLEGAL FORMAT CHARACTERS WERE ACCEPTED AS BLANKS.' or
→ 'THE INTERPRETATION OF MEANINGLESS INPUT WAS ATTEMPTED.'

Record image is displayed. Messages are not repeated if errors recur in same I/O statement. No termination if ERR clause is not specified.

02	00	00	Location of Record

Message: 'RECORDS EXCEEDING MAXIMUM LENGTH ARE FAULTY.'

Record image is displayed. Messages are not repeated if same error occurs in same I/O statement. No termination if ERR clause is not specified.

The six possible error messages produced by NTRAN are:

(1) **NTRAN ERROR* UNIT $: NO PACKET SPACE AVAILABLE.

This message indicates that all available NTRAN packets are in use and that another packet is requested.

Suggested Action: Reassemble NTRAN and increment the number-of-packets-parameter NPKTS.

(2) **NTRAN ERROR* UNIT $ IS NOT AVAILABLE FOR NTRAN.

A reference to a unit was made that is already in use by normal I/O.

Suggested Action: Change unit number.

(3) **NTRAN ERROR* UNIT $ NOT ASSIGNED.

A reference to an unassigned unit was made with a function other than write (1) or assign (19).

Suggested Action: If a write function (1) had been used as the first reference, a dynamic assign of a FASTRAND file (scratch) would have been made. If a scratch file was not intended, an assignment has to be made either by function 19 (assign) or by an assign card.

(4) **NTRAN ERROR* UNIT $ HAS IMPROPER DEVICE

Requested function is not available for the device assigned to this unit. The requested function will be ignored.

Suggested Action: If action is wanted for the requested function, a unit with another device assigned has to be used.

(5) **NTRAN ERROR* UNIT $ HAS ILLEGAL FUNCTION CODE

(6) **NTRAN ERROR* UNIT $: NUMBER OF ARGUMENTS IN STACK EXCEEDS TABLE LENGTH.

This message indicates that the number of arguments in call is greater than the maximum calling sequence table length.

Suggested Action: Reassemble NTRAN and increase the NCT length.

NOTE: The user must not change any argument of an argument group before the function is completed; that is, before the status word (if any) has been changed from −1 to another value. All NTRAN functions are executed in sequence; the completion of one function implies completion, successful or unsuccessful, of all preceding functions.

4.19. NSTAT$ – THE I/O ERROR STATUS WORD

New values for the FORTRAN V I/O error status word have been added for more detailed error analysis during execution of a FORTRAN V program. Contents of NSTAT$ is retrieved by the function reference INSTAT(L) where L is irrelevant. The value of NSTAT$ is set to zero following this reference.

The format of the error word, NSTAT$, is:

S1	S2	S3	H2

S1 always contains the (FORTRAN V I/O) error code. The FLD function may be used to examine the fields of the error word.

The new FORTRAN I/O error routine (element NIOERS) will search for an ERR or END clause. It will also print the error message provided by the calling I/O routine. If the error is associated with format interpretation and/or a record (including a line of NAMELIST I/O), the entire record is displayed following the message:

'THE FOLLOWING RECORD IS ERRONEOUS OR DOES NOT CORRESPOND TO FORMAT SPECIFICATIONS'

If the file is tape, FASTRAND, drum, or an alternate symbiont file, the unit number n is indicated in the message

'*** UNIT n IS IN ERROR'

APPENDIX B

ROUTINE FSESFL - DESCRIPTION & USE

5. FSBSFL--POSITION OVER FILES

5.1 IDENTIFICATION

POSITION OVER FILES.
ELEMENT NAME--FSBSFL.
AVAILABLE IN MSC-LOCALlB.
ADDED IN OCTOBER 1969.

5.2 PURPOSE

THIS ROUTINE WILL POSITION FORWARD OR BACKWARD THE SPECIFIC NUMBER OF FILES ON NON-FORMATTED FORTRAN TAPES. BOTH TAPE MARKS AND SDF END-OF-FILE SENTINALS ARE RECOGNIZED BY THIS ROUTINE.

5.3. USAGE

```
CALL FSBSFL (LUN,NF,NSTAT)
 HERE LUN IS THE FORTRAN LOGICAL UNIT NUMBER
      NF IS THE NUMBER OF FILES TO SKIP.  IF POSITIVE, POSITION
           FORWARD.  IF NEGATIVE, POSITION BACKWARD.
      NSTAT IS THE STATUS CODE.  POSSIBLE VALUES AND CAUSES ARE --
           0=NORMAL RETURN
           1=LOGICAL UNIT IS NOT ASSIGNED TO TAPE OR WAS NEGATIVE
           2=POSITIONED BACKWARD TOO MANY FILES.  THE TAPE IS
           NOW POSITIONED AT THE START OF THE FIRST FILE.
           3=UNRECOVERABLE.
```

5.4. ERROR MESSAGES

NONE

5.5. SPECIAL CONSIDERATIONS

A POSITION BACKWARD REQUEST WILL LEAVE THE TAPE POSITIONED AT THE START OF THE PROPER FILE. IF THE TAPE IS POSITIONED IN FILE 3, A CALL WITH NF = -1 WILL LEAVE THE TAPE POSITIONED AT THE START OF FILE 2.

-158-